DOCUMENT RESUME

ED 070 506 LI 004 061

AUTHOR	Kehr, James E.
TITLE	INTREX Buffer-Controller Display System Operation and Software.
INSTITUTION	Massachusetts Inst. of Tech., Cambridge. Electronic Systems Lab.
SPONS AGENCY	Council on Library Resources, Inc., Washington, D.C.; National Science Foundation, Washington, D.C.
REPORT NO	ESL-R-487
PUB DATE	Sep 72
NOTE	203p.;(0 References)
EDRS PRICE	MF-$0.65 HC-$9.87
DESCRIPTORS	*Computer Programs; Computers; *Information Retrieval; Information Storage; Information Systems; *On Line Systems
IDENTIFIERS	Computer Software; *Project Intrex

ABSTRACT

Intrex is an experimental bibliographic and full text storage and retrieval system which has advanced the concept of the on-line interactive search of a computer stored bibliographic data base and the on-line access to a microfiche document storage. The Intrex system includes a set of computer programs designed for information retrieval, a computer stored data base upon which the retrieval programs act, a microfiche document storage and retrieval system and an assortment of special hardware devices designed to assist the user in his information search. One of the special devices developed for the user is the Intrex Display System which includes a Buffer-Controller incorporating a Varian 620i general purpose computer and a number of remote display terminals. This report deals with the software developed for the Buffer-Controller computer. (Other documents on Project Intrex are available as ED043348, 047739, 053772, 056732, 057829, 060899 and 066207.) (Author/SJ)

ED 070506

U.S. DEPARTMENT OF HEALTH,
EDUCATION & WELFARE
OFFICE OF EDUCATION
THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL OFFICE OF EDUCATION POSITION OR POLICY.

September, 1972

Report ESL-R-487

INTREX BUFFER-CONTROLLER DISPLAY SYSTEM OPERATION AND SOFTWARE

by

James E. Kehr

The research reported herein was made possible through the support extended the Massachusetts Institute of Technology, Project Intrex, at the Electronic Systems Laboratory under a research grant from the Council on Library Resources, Inc., designated as M.I.T. DSR Project number 27808 and through Grant GN774 from the National Science Foundation as M.I.T. DSR Project number 71144.

LI 004 061

FILMED FROM BEST AVAILABLE COPY

Electronic Systems Laboratory
Department of Electrical Engineering
Massachusetts Institute of Technology
Cambridge, Massachusetts 02139

CONTENTS

I. INTRODUCTION TO THE INTREX DISPLAY SYSTEM AND ITS OPERATION page 1

1.1 Introduction 1

1.2 General Description 1

1.3 Intrex Catalog and Text Display System 2

1.4 Buffer Controller Processor 3

1.5 User Operation of the BRISC Terminal 4

1.5.1 Modes of Operation 4

1.5.2 Catalog Mode 5

1.5.3 Edit Mode 9

1.5.4 Text Access Mode 11

II. GENERAL DISCUSSION OF BUFFER-CONTROLLER SOFTWARE 13

2.1 Buffer Controller Software 13

2.2 User Programs 13

2.2.1 Display Console Service 15

2.2.2 Mode of Operation and Function Routines 15

2.2.3 CTSS Message Processing 17

2.2.4 Text Access Transactions 17

2.3 System Programmer Routines 18

2.4 Utility and Special Function Routines 19

III. DETAILS OF BUFFER-CONTROLLER OPERATION AND SOFTWARE 20

3.1 User Parameters 20

3.2 Buffer-Controller Operation 21

3.3 Monitor Program 26

CONTENTS (Cont'd)

	3.3.1	Monitor Initialization	page 26
	3.3.2	Monitor Loop	27
	3.3.3	CTSS Interrupt Service	37
3.4	Remote Login		38
3.5	Remote Initialize		40
IV.	PARAMETER AND SUBROUTINE DESCRIPTIONS		41
4.1	User Parameters		41
	4.1.1	Display Page Parameters	42
	4.1.2	Buffer Address Parameters	44
	4.1.3	Mode Control Parameters	45
	4.1.4	Save Parameters	45
	4.1.5	CTSS Transaction Parameters	46
	4.1.6	Edit Parameters	48
	4.1.7	Text Access Parameters	48
4.2	System Parameters		51
	4.2.1	Console Constants and Variables	51
	4.2.2	CTSS Parameters	53
	4.2.3	Text Access System Parameters	54
4.3	Program Sequence Descriptions		55
	4.3.1	Initialize Monitor Program	55
	4.3.2	Mode Call Entry Points	56
	4.3.3	Monitor Loop	56
	4.3.4	Remote Login	57
	4.3.5	Interrupt Vectors	58
	4.3.6	Utility Entry Points	58
4.4	Subroutine Descriptions		59
	4.4.1	Console Display Service Routines	59
	4.4.2	Drum Routines	65

4.4.3	Line Buffer Routines	page	67
4.4.4	Core Transfers		67
4.4.5	CTSS Message Processing		68
4.4.6	Label Message Processing		70
4.4.7	Mode Change and Mode Entry Routines		70
4.4.8	Edit Routines		74
4.4.9	Text Access Transaction Routines		74
4.4.10	Drum Utility Routines		80
4.4.11	Teletype Routines		81

APPENDICES

A	BRISC TERMINAL FEATURES AND CONTROLS	83
B	MAGNETIC DRUM STORAGE-PRINCIPLES OF OPERATION	94
C	DRUM FILE OF PROTOTYPE DISPLAY LABEL LINES	129
D	INDEX OF FUNCTION CODE TABLES AND INDEX OF ADDRESS TABLES	130
E	FUNCTION ROUTINE LIST WITH CALLING MODES	131
F	FUNCTION, PROGRAMMABLE AND TEXT ACCESS FUNCTION SWITCH CODES	134
G	SYSTEM MESSAGE TRACK CONTENTS	136
H	SOURCE HANDLING PROGRAMS — DESCRIPTION OF THEIR USE	137
I	TEXT ACCESS COMMANDS	145
J	USE OF DRUM TRACK UTILITY	149
K	USE OF OCTAL DUMP UTILITY	150
L	START-UP AND LOGIN PROCEDURES	151
M	REMOTE LOGIN: CTSS-USER TELETYPE DIALOG	157
N	MONITOR PROGRAM FLOW CHARTS	159
O	SOURCE HANDLING CALLING SEQUENCES AND SELECTED FLOW CHARTS	177
P	CORE LOCATIONS AND CROSS-REFERENCES OF ENTRY POINTS AND SUBROUTINES	187

I. INTRODUCTION TO THE INTREX DISPLAY SYSTEM AND ITS OPERATION

1.1 Introduction

Intrex is an experimental bibliographic and full text storage and retrieval system which has advanced the concept of the on-line interactive search of a computer stored bibliographic data base and the on-line access to a microfiche document storage.

The Intrex system includes a set of computer programs designed for information retrieval, a computer stored data base upon which the retrieval programs act, a microfiche document storage and retrieval system and an assortment of special hardware devices designed to assist the user in his information search. The Intrex staff includes information analysts, librarians, catalogers, system programmers and engineers.

One of the special devices developed for the user is the Intrex Display System which includes a Buffer-Controller incorporating a Varian 620i general purpose computer and a number of remote display terminals. This report deals with the software developed for the Buffer-Controller computer.

1.2 General Description

The Intrex retrieval programs were written on the compatible time-shared system (CTSS) operating on the IBM 7094 located at the M.I.T. Information Processing Center. The data base consisting of the Intrex Augmented Catalog generated by a staff of Intrex catalogers and librarians resides on disk storage on CTSS. The catalog information was extracted by this staff from selected journals and conference proceedings in the broad subject area of material sciences and engineering. The Intrex user may conduct information searches upon the data base from any terminal capable of interfacing with CTSS. The device in most widespread use is the IBM 2741 typewriter terminal. Since the output streams from Intrex retrieval searches are often of textual material the speed and quiet operation of an electronic display terminal is helpful in handling the volume inherent in such material. One electronic display terminal, the

ARDS (Advanced Remote Display System) is available and has been adapted for Intrex use. In addition, an alphanumeric refreshed CRT display console was developed for the Intrex Display System and has been incorporated into a terminal called the BRISC (Buffered Remote Interactive Search Console). The BRISC terminal includes a second CRT, a storage-tube display for the presentation of microfiche document images. Currently there are two BRISC terminals in the Intrex Display System.

The Text Access system is a microfiche document storage and retrieval system which incorporates two modified Houston-Fearless microfiche storage, retrieval and display devices. The user may view the full text of a document at either the modified ARDS terminal or on the storage-tube display of the BRISC.

A 35 mm film strip copy of the microfiche document may be ordered from any Text Terminal or from the BRISC. The film strip may be viewed directly in a microfilm viewer or may be used in a modified electrostatic copier to produce a full size 8-1/2 x 11 paper copy of the document.

1.3 Intrex Catalog and Text Display System

The Intrex Display System features the BRISC terminal which includes a refreshed CRT alphanumeric catalog display and a storage tube full text display.

The Intrex Catalog Display is a general-purpose display station designed with special consideration of user requirements for interactive bibliographic search. The display interfaces with a Buffer-Controller which serves to buffer information and control elements of the Intrex display system. Major units of the Buffer-Controller are a small general-purpose data processor, the Varian Data Machines 620i and a Vermont Research magnetic drum unit.

The BRISC incorporates a keyboard and a set of function switches for inputting requests and commands to the system and a refreshed CRT and a set of status lights for display of system output. The magnetic drum provides the refresh memory for the CRT as well as program and message storage for the 620i processor of the Buffer-Controller. The drum also allows local storage of information obtained from the CTSS retrieval programs. Each user is allocated a block of drum tracks for

storing pages of information he wishes to retain locally. He may then view and edit those pages without requiring access to the central computer system. Using the drum for CRT refresh and local storage of display pages greatly reduces the demands on the central processor.

The Buffer-Controller also sends commands to and receives responses from the microfiche storage system. Commands for display on the text display station are queued, formatted and relayed to the Text Access system by the Buffer-Controller. These commands may be initiated by manual operation of function switches on the text display station, directly from the BRISC keyboard, or from Intrex retrieval programs.

The Buffer-Controller is designed to handle a cluster of BRISC terminals. Thus the cost of the local processor and magnetic drum may be shared by multiple consoles.

The console design includes other features useful to the Intrex user. Programmable function switches with labels on the CRT present options available to the user as his search activities progress. A special cursor enables the user to point at items on the CRT screen and a set of buttons enable him to position the cursor rapidly.

1.4 Buffer-Controller Processor

The Buffer-Controller incorporates a Varian Data Machine's 620i minicomputer.

The 620i is a general purpose digital computer with 8192 (8k) 16-bit words of core and a 1.8μs cycle time. It has an extensive instruction set and flexible input-output design with priority interrupt and direct memory access. Arithmetic is done in two's complement notation. The order code includes both single and double operand instructions. Addressing used is direct, relative to program counter, relative to two index registers and indirect. An ASR-33B Teletype keyboard and printer with mechanical paper tape reader and punch serve as the basic system I/O device.

A special Drum Control Unit provides the interface for the Vermont Research Drum to the 620i I/O logic. The current drum supplies 128 data tracks and 6 timing tracks. Each track holds 1024 sixteen bit words (or 2048 eight bit characters). Each track corresponds to a

single page of alphanumeric information on the Intrex refreshed CRT display.

The 620i receives characters from the keyboard of the BRISC and writes them on the drum track which is currently refreshing the CRT display. The 620i also presents messages generated by the local processor and those received from the central computer. Once the alphanumeric data is written on the drum the display system hardware automatically refreshes the CRT. The refresh operation is suspended only long enough to permit new data to be written on the display track.

In addition to servicing the multiple BRISC terminals the 620i controls data flow between the various other devices of the Intrex Display System. It regulates the user communication with CTSS via dataphone and telephone line and relays system commands and responses between user and the Text Access system. Each dataphone to CTSS operates in half-duplex mode, receiving data at 1200 baud and transmitting it at 110 baud.

User actions at the BRISC and current system conditions are sensed by the 620i Monitor Program which then calls into operation the appropriate user programs to initiate and control the sequences of required machine response.

1.5 User Operation of the BRISC Terminal

1.5.1 Modes of Operation

The Intrex Display System has a large set of realized and potential capabilities. To simplify the user operation of the BRISC terminal and to provide a convenient organization of 620i software the concept of mode of operation was adopted. The current system employs four distinct modes, Catalog, Edit, Text Access and Source Handling. The primary user transactions of bibliographic search and full text retrieval are done in Catalog Mode. The Edit Mode allows the user to alter the contents of his display pages by editing and annotating their contents. The Text Access Mode allows the user to enter fiche access numbers and obtain full text display without calling upon CTSS. This is used primarily for system testing and demonstrations of the microfiche display system. The Source Handling Mode is currently for use by system programmers only and cannot be entered from the BRISC terminal without operator intervention at the

620i control panel. However, many of the Source Handling Routines have general data manipulation functions and could be made available to the sophisticated user.

1.5.2 Catalog Mode

1.5.2.1 Search Dialog Display

The BRISC terminal was designed primarily to allow the user to conduct an interactive search of the Intrex data base and to obtain display of the full text from the microfiche document storage. These functions are available in what is called the Catalog Mode of operation of the BRISC terminal. The user types on the keyboard commands to the Intrex retrieval programs which reside in the CTSS time shared system. The 620i of the Buffer-Controller receives the typed characters from the BRISC keyboard and relays them to CTSS via a data-phone and telephone line link to the 7094 at the M.I.T. Information Processing Center. CTSS responses are received by the Buffer-Controller over the same link and are buffered and displayed for the user on the refreshed CRT screen of the BRISC. When the screen has been filled with the user-machine dialog the user may select another fresh page by use of the New Page function switch on the BRISC. When the user presses New Page, the previous page of dialog is automatically saved in the local drum storage.

The user may look back at a previous page of dialog by pressing the Previous Page function switch. Return to his current dialog page is made by pressing the Next Page function switch. Pages are numbered in the lower right corner. The user has 14 pages of drum storage for use in conducting his search dialog. He pages forward with the Next Page function switch and backwards with the Previous Page function switch. He can view Page 1 by pressing the First Page button. Pressing Next Page when on Page 14 causes Page 1 to be displayed. Pressing Previous Page when on Page 1 causes Page 14 to be displayed.

The user should not type commands or press function switches unless it is his turn in the interactive dialog. CTSS ends its transmission with a READY message to indicate that it is the user's turn. The user may then employ the BRISC function switches or type Intrex commands to the retrieval system. The user terminates the commands with a

carriage return. It is then the system's turn to act until the next ready is displayed or until an end of page condition (see below) occurs. The user may interrupt CTSS output, however, by pressing the Attention/ Quit button on the terminal.

If the user types commands when it is not his turn in the dialog, the typed commands may appear on the CRT screen but will be ignored by CTSS. If the user presses the paging function switches when CTSS is transmitting, the CTSS message will appear on the currently visible page on the same line that would have been used on the original dialog page. This may result in the new message writing over old display information.

When the display page becomes full and CTSS has not completed the response to the last user command, an end of page condition exists at the BRISC. A special character ␣ sent by CTSS is displayed and an End of Page status light appears on the keyboard. At that time CTSS waits for the user to take the next action before resuming the unfinished transmission. Most frequently the user will press the New Page button which will cause the 620i to select a new display page, initialize it and signal CTSS to continue at the top of the new page. However, the BRISC allows the user other options. He may decide not to save his current display page and so may press the Erase programmable function switch to clear the current page and signal CTSS to continue with the unfinished message at the top of the current display page. In fact the user may employ the Next, Previous, or First Page buttons to select any one of his 14 display pages and clear it with the Erase button to receive the unfinished message.

When in Catalog Mode the last line of the CRT refresh display indicates the current display page number and also presents labels for the four programmable function switches on the left. The left-most switch is the ERASE button used to clear the display page being viewed. The use of the other three switches is described below in the section Note Page.

The last display line which is called the label line, is also used to present special status messages from the Buffer-Controller to the user. For example, if the user is in Catalog Mode, his current display page is 14, and he presses the New Page button, the 620i will select page 1 for display and will use the label line to inform the user that

he has reached page 1 again and must press Erase (or New Page) to continue his dialog. The next keyboard action will cause the normal catalog label line to appear. When the user does proceed from page 14 to page 1 in Catalog Mode it is expected that he will inspect that page for any information that he wishes to retain and take notes either by paper and pencil or by use of the Note feature described below, before pressing Erase to continue. Use of Erase removes that page of display information from the local storage. If the user continues on by pressing the New Page button, he automatically erases each page selected and prepares it to receive new information. Warning is given to the user only when he makes the transition from page 14 to page 1.

1.5.2.2 Note Page

In Catalog Mode lines of displayed information may be retained on a user's special note page. This page is distinct from the user's normal 14 display pages. Three programmable switches labeled NOTE, NCLEAR (Note Clear), and NVIEW (Note View) are associated with the note-taking procedure.

In order to prepare for note-taking the user presses the NVIEW button to see what is currently written on the Note Page. If it is something which he does not wish to retain he may obtain a blank page by pressing the ERASE button. He then returns to his display page by pressing the RETURN button and presses the NCLEAR button to indicate that his first noted line is to be placed at the top of the Note Page.

Having prepared the Note Page the user may now save lines of displayed information there. In order to transfer a line to the note page he first positions the blinking cursor symbol on the line. The cursor control buttons TOP, RIGHT, LEFT, UP, and DOWN are used to position the cursor. Pressing and then releasing the button causes the cursor symbol to move one step. Holding the button down causes the cursor symbol to advance in steps until the button is released. A Home button places the cursor symbol at the top left character position on the CRT screen.

After the cursor is in position the user presses the NOTE button. A slight blink of the display indicates that the note was taken. He may then move the cursor to another line, press Note and so continue until up to 30 lines of information have been saved on the Note Page. At any time

during the procedure he may view the Note Page by pressing the NVIEW button and then return to his current display page by pressing the RETURN button.

The purpose of the Note Page is to allow the user to condense information from his 14 display pages and so permit him to erase and re-use those pages for his continuing search dialog. If the Note Page becomes full the user may employ the Edit Mode features of the terminal to move the noted information to another page and free the Note Page for erasing, clearing and further note-taking. The Edit Mode is described below.

1.5.2.3 Microfiche Text Display

While in Catalog Mode the BRISC user may type the Intrex command, output text. The first time it receives this command CTSS will respond by sending the fiche information to the Buffer-Controller for the first page of the first document on the users current search list. Successive consecutive uses of the output text command will cause the fiche information for the first page of successive documents on the user's list to be sent to the Buffer-Controller. The Buffer-Controller displays the fiche information on the CRT and then processes the text display request so that the image of the first page of the microfiche document will appear on the storage-tube display screen.

Associated with the text display screen is a set of lighted function switches which are called the Text Access function switches. Switches marked First Page, Next Page, Previous Page and Last Page are used to view the pages of the microfiche document. The Text Access function switches are active in both Catalog and Text Access Modes.

Attempts to use Previous Page when on the first page, or Next Page when on the last page causes the Buffer-Controller to place warning messages on the refresh CRT label line. Under this circumstance the light buttons will go out when pressed and not be relighted until a valid text request is made and is completed. Whenever a valid request is made a Handshake message on the refresh CRT label line informs the user that contact has been made with the text system. The speed of response depends upon the number of users accessing the microfiche storage at that time. At the completion of the text transaction text status inforamtion

is presented on the refresh CRT label line. Messages such as Text Access System Ready, Fiche Not Found — Ready, Text System Malfunction — Ready, inform the user of the results of his last request. A set of status lights on the text display station convey some of the same information.

The fiche page image may be enlarged by use of the magnify sector function buttons on the text display station of the BRISC. Each sector corresponds to one quarter of the page. The position of the sector on the page is indicated by an array of 9 sector buttons. The location of the sector button in the array corresponds to the location of the sector on the page to be magnified. The Same Page function button may be used to obtain the full page image again after using a magnify sector button. The Same Page may be used at any time to reselect a fiche. The Same Page function is useful when adjusting the image intensity. The storage-tube intensity must be adjusted before the image is written, or while it is writing on the screen. This may call for several uses of the Same Page button before the adjustment is satisfactory. The Same Page button may also be used if the page image is not properly centered when first received.

1.5.3 Edit Mode

The user conducts his bibliographic search and obtains microfiche document display while operating in the Catalog Mode. He may take notes on his Note Page and view that page without leaving Catalog Mode. For the user who is more familiar with the BRISC and the Intrex system, a special Edit Mode is provided so that the user may edit the material he has accumulated on his 14 display pages and on his Note Page. The editing is done within the Buffer-Controller and does not affect, nor require, the operation of CTSS nor the Text Access system.

Entering the Edit Mode places the terminal in the Cursor Mode of operation. To replace any character on the screen, the user positions the cursor symbol on the character to be changed by use of the cursor control buttons and then strikes the key of the desired replacement character. The cursor symbol moves to the next typing position as each key is struck.

In Edit Mode the programmable function switches present the user with special editing functions. The cursor is used to indicate any line the user wishes to alter. Pressing the programmable function switch labeled ERLNE (Erase Line) causes the program to erase the line on which the cursor lies. Actuation of the EROP (Erase Rest of Page) switch erases the line on which the cursor lies and all the material that follows on that page except the words that constitute the labels for the programmable switches.

The SVLNE (Save Line) programmable switch causes the line indicated by the cursor to be saved on a drum track assigned for that purpose. Pressing the RTLNE (Return Line) switch causes the saved line to appear on the display at the line position indicated by the cursor. The user may save up to 30 lines by positioning the cursor and pressing SVLNE for each line. He may then return those lines to any line on any page by selecting the page and the line and pressing RTLNE once for each line that was saved. The lines are returned in the same order in which they were saved, so the first line saved is the first line returned.

The page on which lines are stored by use of SVLNE is the same as that used by the Note feature of Catalog Mode. The SCLEAR (Save Clear) switch resets the save line and return line mechanism so that after pressing SCLEAR the next operation of SVLNE or RTLNE will operate on the first line of the Save Page. Thus normally the user will press SCLEAR before his first use of SVLNE. He may view the contents of the Save Page by pressing SVIEW (Save View).

As the user conducts his bibliographic search in Catalog Mode he may use the Note feature to take notes as the search progresses. If he fills the 30 lines of the Note Page and has not finished his search, he may enter Edit Mode and move the 30 lines to one of his fourteen display pages, press the SCLEAR button to reset the Save mechanism to the top of the Save Page, press Erase while viewing the Save Page to clear it of old information and then return to Catalog Mode to continue his search and note-taking.

A proposed use of the Note and Edit features is that the user may construct his own bibliography from catalog output. He may supplement and reformat the obtained information and then return it to CTSS

for either an off-line printer output or an on-line typewriter terminal output. The ability to return the information from a BRISC display page to CTSS for hard-copy output is not implemented in the current system.

1.5.4 Text Access Mode

The Text Access Mode is entered when the user wishes to access the microfiche storage and display system without employing the Intrex catalog retrieval programs of CTSS. In order to do this the user must know the fiche access number for the document whose text he wishes to view. It is also a useful mode of operation when a system check-out is being made by system programmer/operators.

The Text Access Mode is entered by pressing the Text Mode function switch on the BRISC keyboard. The user may then type the fiche access number in the form

FFFF RCC NN(CR)

where

FFFF	is a 1 to 4 digit fiche access number
R	is an alphabetic character between A and E (inclusive) indicating the row of the first page frame
CC	is a 1 or 2 digit column number of the first page frame
NN	is a 1 or 2 digit number indicating the number of pages (frames) in the document
(CR)	is a carriage return.

The Intrex correction characters of # for delete the last typed character, and @ for delete the current line are active in Text Access Mode.

All the function switches on the Text Display station are active in the Text Access Mode as well as in Catalog Mode. The Text Access Mode has a label line which places labels on the four programmable function switches on the right side of the refreshed CRT. These switches are currently labeled TCOPY (Text Copy), TFIRST (Text First Page), TLAST (Text Last Page), and TRESET (Text Reset). The TFIRST and TLAST functions are duplicates of the First Page and Last Page switches on the Text Display

Station. They allow testing of the Buffer-Controller/Text Access interface without use of the Text Display Station Switches.

The TCOPY (Text Copy) switch causes a request for a 35 mm film strip copy to be made of the last document the user has had displayed. The user first makes a display request for a microfiche document and may then press the TCOPY button if he wishes a film copy of that document.

The TRESET (Text Reset) button may be used if a text request has been made and after a two minute wait no text response has occurred. The Text Reset allows the Buffer-Controller to issue another text request even though there was no response from the last request. This is a 620i software reset function and it usually does not correct the reason for the previous non-responsiveness of the system. It is largely for the use of the system programmer/operator in diagnosing malfunctions.

The above four programmable switches are also active in Catalog Mode; however, they are not labeled in Catalog Mode to simplify the appearance of the display in that Mode. They will not be employed by the normal Intrex user. The Text Mode is largely for use by the system programmer/operator. It may be used to demonstrate to visitors the functioning of the Text System when CTSS is not available.

II. GENERAL DISCUSSION OF BUFFER-CONTROLLER SOFTWARE

2.1 Buffer-Controller Software

The 620i Buffer-Controller software falls into three general categories: the operating user programs associated with the BRISC terminal; the system-programmer routines used in generating and handling 620i source programs from the BRISC; and the utility programs for special functions such as core dumps, listings, loading, and assembly, all operating from the ASR 33-B teletype terminal.

2.2 User Programs

The user programs are accessed from an operating monitor program (Monitor). User actions at the BRISC terminals are monitored and keyboard entries initiate the execution of user function routines. Incoming messages from CTSS interrupt user program execution and briefly take control from the Monitor long enough to transfer characters received from the CTSS hardware data phone interface buffers to 620i core storage for later processing. Then control is returned to the user programs and the Monitor. Input sources other than CTSS set hardware flags that are monitored and processed in a fixed order.

The monitor program has three kinds of duties as shown in Fig. 2.1. First it takes and processes input from the user display consoles. Some user initiated functions are taken to completion immediately while others are merely started and must be completed at a later time. Second, the monitor program advances the processing of any buffered input from CTSS. Third, the monitor advances pending transactions with the microfiche text access system.

Since the monitor program must deal with independent on-line users and with system devices with highly variable timing requirements, many functions are broken down into separate parts and the execution of the parts of one function are interlaced with the execution of parts of other functions. Such an organization increases the likelihood that all inputs to the system are fully monitored. It of course adds complexity to program control.

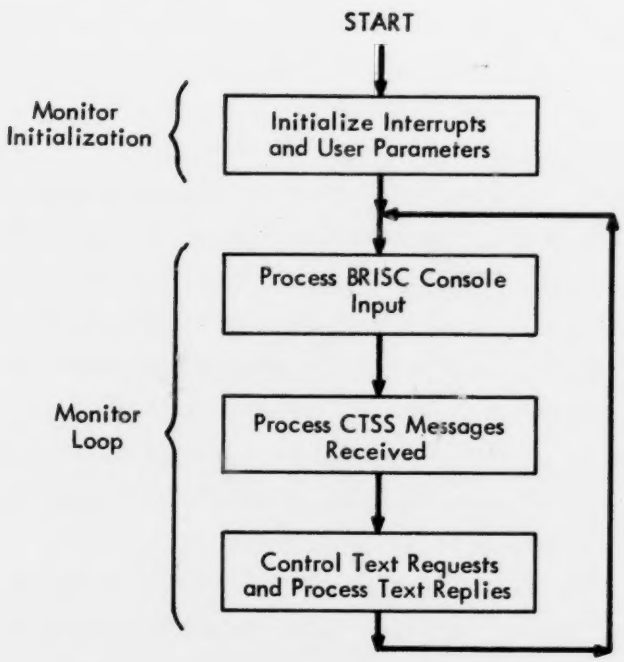


Fig. 2.1 Monitor Initialization and Loop

2.2.1 Display Console Service

The monitor program first checks to determine if any of the BRISC terminals are requesting service. Terminals have an assigned priority which determines the order of service. A keyboard input program is called to interrogate the highest priority console requesting service. The user console responds with a 40-bit reply to the console hardware interface buffer. The console input is processed and results in either the execution of a function routine corresponding to the use of a console function switch, the display of a typed character on the refreshed CRT, or no action at all in the event of an invalid input. The action taken depends upon the status of the user console being serviced. Associated with each user is a list of software parameters which fully define the status of his console. These parameters indicate the working mode of the console and point to the tables of function codes and function entry points, and the display label lines appropriate for that mode. The user parameters also contain information pertaining to the user's CRT display page.

Character keys typed on the BRISC keyboard generate character codes which are written at the correct drum character position according to the current user display page parameters, so that they appear in the appropriate position on the user's refreshed CRT display. The BRISC hardware converts the binary character codes stored on the drum into the dot matrix light display representation presented to the user on the CRT screen.

Characters received from a BRISC keyboard are inspected and certain special control characters such as the Intrex correction characters, @, #, and backspace, initiate function programs associated with the CRT display. Other characters such as the carriage return initiate functions associated with the display page and with the console mode.

2.2.2 Mode of Operation and Function Routines

The Monitor receives and interprets user actions at the BRISC console. The response to user inputs is regulated by the user-console status information retained by the 620i. There is associated with the user-console over 60 user parameters. One of these parameters is called MODE. The console MODE determines how the user input will be treated.

Normally the user operates in the Catalog Mode which allows him to conduct a dialog with CTSS. Characters typed on the BRISC keyboard are sent to CTSS and messages received from CTSS are presented to the user on the CRT refreshed display screen.

Two other modes of operation available to the user are the Edit and Text Access modes. In these modes characters typed on the BRISC keyboard are not sent to CTSS and messages received from CTSS are buffered in the 620i and not displayed on the CRT until the Catalog Mode is re-entered.

In the Edit mode the user is dealing only with the Buffer-Controller and the associated 620i and magnetic drum. He has a special set of options available through the BRISC programmable switches whose functions are labelled on the last line of the refreshed CRT display page. These options allow him to perform operations on the content of the display pages assigned to his console. The CTSS program is not aware of any user action or its consequences which occur when the user is in either the Edit or Text Access mode of operation.

While in Catalog Mode the user may make normal text display requests. The Text Access function switches are active in Catalog Mode. The necessary fiche information for such text requests are derived from the current user text parameters. The Text Access mode is provided so that the user may employ the BRISC as a Stand-Alone Text Station. In Text Mode characters typed at the keyboard are not sent to CTSS but are displayed on the CRT refreshed display page and when a carriage return is encountered those characters typed since the last carriage return are analyzed and converted into fiche information for a text display request. Error detection routines are employed to insure a valid text request. The Intrex kill characters @, #, and backspace are available in Text Mode to correct typing errors.

Text Access display and film copy requests may be initiated by use of function switches whose use is detected by the monitor program as it services the display consoles. However, the requests are only formatted and placed in the text request queues at that time. The actual transactions with the Text Access system are conducted later in the monitor sequence.

2.2.3 CTSS Message Processing

Each CTSS input to the Buffer-Controller for a BRISC terminal generates a 620i processor interrupt which takes control from the monitor program. Each user has a separate interrupt address and the interrupt service routine places the character received in an unpacked user character buffer reserved for that user. Control is returned to the Monitor and,after it has completed its current slice of service to a BRISC console, it then moves unprocessed CTSS characters from the user character buffers to user line buffers. Each pass through the monitor loop moves one character for each user, if such characters are waiting in the character buffers.

When a line is complete the Monitor executes a drum line write operation to transfer the received characters from the line buffer in core to the drum track which refreshes the user's CRT display. The user parameters are accessed to determine the correct position on the screen and then the user parameters are updated for the next drum write operation. The drum line write operation makes use of the 620i Direct Memory Access feature and so the Monitor merely starts the write operation which is completed by cycle-stealing from the CPU.

2.2.4 Text Access Transactions

The text access storage and retrieval system is shared by various terminals and systems. The BRISC consoles and the Buffer-Controller is only one of such systems. The Buffer-Controller makes requests to the Text Access Central Station for all BRISC consoles via a single Text Access Interface. Two kinds of responses are returned to the BRISC terminal. The text page image is transmitted as a video signal directly to the Text Display Station of the BRISC. In addition special Text System Replies come to the Buffer-Controller as a four-bit coded message. These coded replies convey status information about the last text request made through the text interface. Such coded messages as Fiche Found-Ready, Fiche Not Found-Ready, Handshake-Proceed and others are sent from the Text Central Station.

Since the text system is called by devices not under direction of the Buffer-Controller, the monitor program does not have information about the status of the text system at a given time. The monitor program

of the Buffer-Controller outputs text request data to the text access interface when any BRISC user makes a text request, if the interface is not in current use. The text access interface hardware then handles the procedure and timing requirements of the transaction. The monitor program is supplied with status information about the progress of the transaction so that further text requests may either be initiated immediately or delayed.

Text requests generated from the BRISC terminal are of two types, image display and film copy. Display requests have priority over film requests because the display requests engage the text access system for less time and because the film copy process is a remote operation.

The Monitor has separate queueing mechanisms for the text display and the text film requests. Each user can have only one request in the display queue. He makes a new request when he is ready to view a new page or a new document. The viewing is truly an on-line process while film requests are merely on-line requests that film be made at a remote location. Since the user receives no on-line response to the film request, the film queue carries no user identification. The image display does produce an on-line response to the BRISC user and so the display queue must contain user identification. This is done by having a separate display request buffer in core for each user. The collection of these buffers constitutes the display queue.

The Monitor determines the next display request to be processed while the current request is still in process. Thus one user cannot monopolize the text system.

2.3 System Programmer Routines

A set of programs was developed to assist the system programmer in generating source programs for assembly. Source statements may be typed at the BRISC terminal onto the user display page drum tracks thus creating source files. The use of the programmable function switches then allows the user to manipulate the text of those files. He can cause the source statements to be packed as a continuous string on display tracks and can unpack them to make corrections and changes. He can transfer source files from one set of tracks to another for

storage. Input to display tracks can be transferred over to system tracks for storage. The contents of drum tracks can be output onto paper-tape for back-up storage, and the paper-tape can be read in for drum storage and display page presentation. In addition, the editing features of the Edit Mode are available for use in manipulating source files.

These functions could be of use to a sophisticated Intrex Catalog user; however, controls and limits should be placed on the programs before they are made available to the general user so that system files are not endangered.

In the current system the 620i operator must start-up at the Source Handling Entry Point to make these routines available at the BRISC terminal. This action places the terminal in the Source Handling Mode.

2.4 Utility and Special Function Routines

A set of general utility programs are available for use by the system programmer/operator at the 620i front panel and the ASR 33-B teletype unit. They allow the programmer to make octal dumps of core contents on the teletype (TTY) printer, or to make reloadable binary dumps onto paper tape and to reload them.

A set of drum utilities allows core images to be written onto drum tracks and track contents read into core from the 620i front panel.

The 620i Assembly Program supplied by Varian Data Machines has been modified to accept its source statements from drum files instead of paper-tape files and to build its symbol table on drum tracks instead of in core. This allows the 4K Assembler symbol tables essentially unlimited size. The modified assembly program is stored as a drum file on the allotted system drum tracks.

III. DETAILS OF BUFFER-CONTROLLER OPERATION AND SOFTWARE

3.1 User Parameters

There is associated with each user at a BRISC terminal a set of software parameters which define that user's operating status for the monitor program. Some of the parameters refer to the detailed condition of the user's CRT display page. The drum track, line and character position at which the next user typed character is to be displayed, or the user's current display page number are examples of such user display parameters. Other parameters deal with the user's text access system requests. These include the fiche number, frame row and column, number of pages in the document and other information about the last text request made by the user. A more complete discussion of user parameters is given in Chapter 4.

One of the user parameters indicates the mode in which a given console is operating. The usual operating mode for Intrex users is Catalog. In that operating mode messages from CTSS for that user are processed and displayed on the user's current display page and characters typed by the user are sent over the CTSS data-phone to CTSS.

The bottom line of each display page is set aside for a label line containing the abbreviated names for the programmable function switches and for special status information such as the current user page number, and the codes to light the user's console status lights. Each operating mode has a table of function codes which determines the user functions available in that mode. For modes other than Catalog the label line corresponding to the mode is stored on the drum track LBTK (Label Track). The numerical value of Mode is the line number of that mode's label message on the label track. The label line for Catalog mode remains in core since it is the most commonly used mode and reading the label messages from the label track delays the monitor program.

Also associated with each mode is a function code table and function routine entry or address table. The user presses a function switch which generates an eight bit coded input to the 620i. The routine KBI (Keyboard Input) extracts that code and calls FSDE (Function Switch

Decode) to decode the request and execute the corresponding function routine. The code table for a given Mode contains all the eight bit function codes corresponding to the functions which can be called from that user mode. The codes are packed 2 per 16 bit 620i word. FSDE searches the code table corresponding to the user mode for a match with the input code. If no match occurs the input is ignored. Items in the second table, the function routine entry table, for a given mode, correspond by position to items in the code table. Thus the first item in the code table for a given mode corresponds to the first item in the address table for that mode. Therefore finding a match in the code table locates a function routine entry point. Functions can be disabled or removed by simply placing a zero in the corresponding address table without the necessity of reordering the code table.

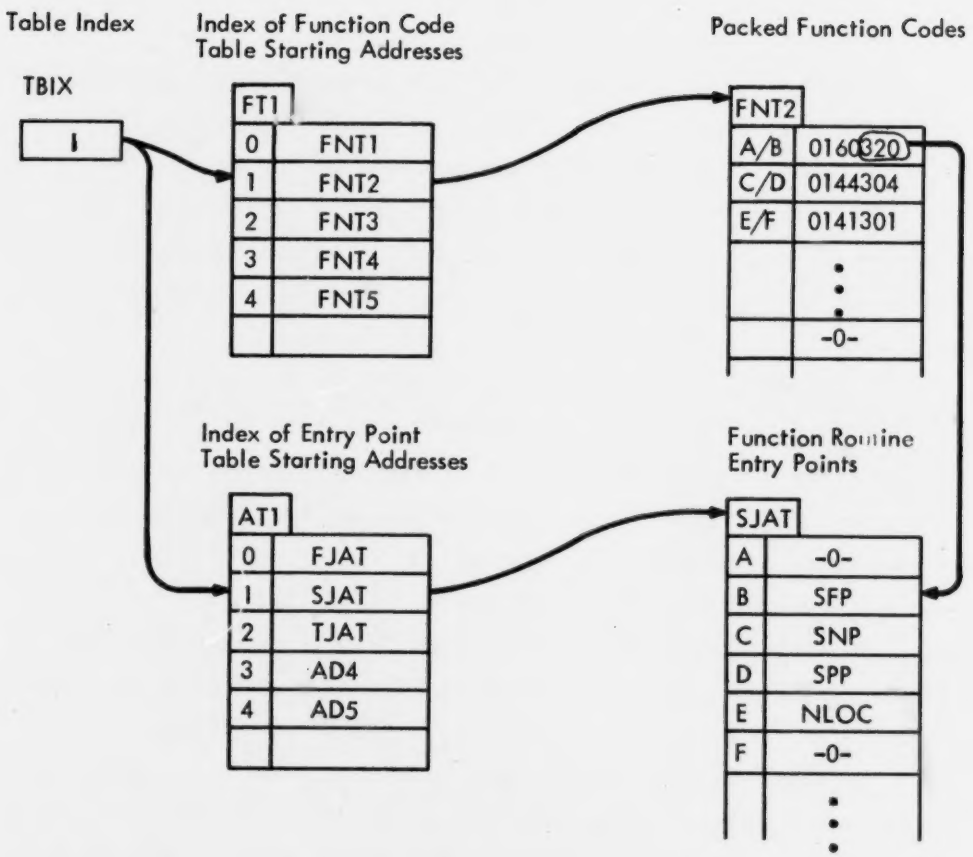
Whenever a mode of operation is entered the user parameters MODE and TBIX are given values corresponding to that mode. TBIX (Table Index) is an index number that is used to locate the table of function codes and the table of function routine entry points to be used in decoding user function requests.

In the current system TBIX takes on the values 0 to 4. The routine FSDE (Function Switch Decode) uses TBIX to locate a pointer in an index of Function Code Table Starting Addresses and also to locate a pointer in an index of Entry Point Table Starting Addresses.

As shown in Fig. 3.1, if TBIX has a value of 1, it points to the item FNT2 in the Index of Function Code Table Starting Addresses and also points to SJAT in the Index of Entry Point Table Starting Addresses. FNT2 is the starting address of a table of function codes which are active for any mode which has a TBIX value of 1 and SJAT is the starting address of a corresponding table of function routine entry points.

3.2 Buffer-Controller Operation

Most often the Buffer-Controller operates in Catalog mode. Catalog mode allows commands to be issued to the Intrex retrieval programs residing in the time shared system (CTSS). A catalog search may be conducted with the Buffer-Controller controlling and displaying the resulting on-line dialog. The buffer drum storage allows the user to retain up to fourteen display



Example: Edit Mode user call for First Page Function Routine.

Fig. 3.1 Table Index Pointers for Interpreting Function Codes

pages of such dialog as it develops. When a current display page becomes full the user presses the NEW PAGE function switch to obtain a new blank page in order to continue the dialog.

Both CTSS and the 620i maintain a line count for the displayed dialog. As part of the login procedure, a CTSS system parameter was set to regulate BRISC page display to 30 lines with a capacity of 55 characters per line. If the CTSS line count exceeds 30 while CTSS is transmitting, CTSS halts the transmission after sending a DELETE character (0177 octal) before the CR (Carriage Return) on the 30th line. In the 620i either the receipt of the DELETE character from CTSS or the attempted advancement of the line count to 31 will generate an End of Page condition. CTSS will not make further transmissions (with a few exceptions) until the Buffer-Controller signals that it is ready for additional input by sending the Form Feed (FF-octal 014) character to CTSS. The 620i will not send the FF until the End of Page condition is cleared at the BRISC console. The user action necessary to clear thus condition is described below.

When the End of Page condition occurs the monitor program causes a control character (octal 021) to be written on the display page label line. The BRISC hardware detects this code in the CRT refresh data coming from the drum and lights the End of Page status light on the BRISC terminal. The DELETE character if it has been sent by CTSS is also displayed as a special character ␣ on line 30.

Under certain circumstances the End of Page condition may be generated when the line count is less than 30. Use of the Attention/Quit button to generate interrupts to CTSS sometimes causes this to occur. Such an End of Page condition must be cleared before dialog can be resumed.

It is up to the user to take some action at the BRISC to clear the End of Page condition. This may be done in several ways all of which cause a software call to the routine ERASE. The usual action is to press the New Page function button. The monitor program then selects a new drum track, clears it of old information, establishes a new label line and page number, and updates the console status light flags and other program parameters to create a top-of-page condition. The 620i also transmits a FORM FEED character to CTSS to permit continuation of output

which is then displayed on the new page. Pressing the NEW PAGE function switch automatically preserves the user's old display page in the Buffer-Controller by the selection of a new drum track for the storage of up-coming dialog. The user has access to fourteen pages of buffer storage. If he presses NEW PAGE when his current page is Page 14, the 620i selects the drum track corresponding to the user's Page 1, but does not initialize that track nor set up the top-of-page condition. Instead it places a message on the display label line (last line of the display screen) to inform the user that he has recycled to Page 1, and must use the ERASE function switch to continue.

The system also allows independent use of the NEXT PAGE and ERASE function switches so that the experienced user may select any of the fourteen pages to serve as his current display page. Thus after recycling to Page 1 he may choose not to erase pages with important information but instead to resume output on a page of his choice.

Alternatively the user may clear the End of Page condition by merely pressing the ERASE function switch. In this case he causes the contents of his current display page to be cleared and the dialog may continue on his current page. Thus no local storage of the previous dialog page is maintained.

Execution of the ERASE routine (either directly by pressing the Erase programmable switch or indirectly by use of the New Page function switch) causes the buffer-controller to transmit a FORM FEED character to CTSS. At this point both systems are in page synchronization at the first character position of the top line. From this time both CTSS and the 620i count carriage returns originating from either system to determine the current line number. A call to ERASE at any time resynchronizes both systems at the top of page.

The BRISC is designed with the flexibility that allows the user to alter his typing position by a deliberate use of the cursor mode of operation. Such action may cause a loss of page synchronization and CTSS may indicate an end of page condition before the page is filled. Under those circumstances CTSS transmits the DELETE character which the monitor program detects thereby creating an End of Page condition in the Buffer-Controller software.

In the event the monitor program has filled the last display line and CTSS continues to send messages, the current system simply reuses the last display line for the additional output from CTSS. Such overwriting of information will continue until CTSS ceases to transmit. It is expected that the user will then employ the NEW PAGE or the ERASE button to reestablish page synchronization.

One of the techniques used to reduce the above problem is to preserve user display parameters when mode changes are made. When the user goes from Catalog to Edit his display parameters are saved and he automatically enters Cursor mode. When he returns to Catalog mode his display parameters are restored to the values held when he left Catalog mode. Thus his typing position on the page will be the same as it was when he left Catalog Mode. This technique is not used when going from Catalog to Text Access mode because it will probably lead to the user overtyping his fiche information when he returns to Catalog mode. This was considered more undesirable than the risk of losing page synchronization with CTSS. In practice this is not a significant problem since the normal user will not alternate operation between Catalog and Text Access mode. Use of the NEW PAGE or the ERASE button should always restore page synchronization.

In the current hardware the position of the cursor is not under 620i program control, but rather is controlled by the user who moves it by use of the cursor control buttons or by use of the keyboard keys. As he types the cursor moves along the screen and is positioned properly for Return, Line Feed, New Page, Back-Space, etc. In Catalog Mode the dialog furnished by CTSS uses character positions and lines on the screen, but since the cursor is not under program control the cursor position remains where the user left it after his last keyboard action. For this reason when in Catalog Mode the 620i does not use the cursor position to determine where the user's typed characters go on the screen. When the user enters Catalog Mode the 620i software places the console in a Non-Cursor Mode. Only by pressing the CURSOR MODE function switch or by entering Edit Mode does the console get into the Cursor Mode of operation. In Edit Mode the cursor position and the typing position correspond and no CTSS dialog is permitted. Returning from Edit Mode to Catalog Mode puts the console in Non-Cursor Mode.

The Note feature requires the Cursor symbol while at the same time the catalog dialog requires a Non-Cursor operation. Thus when in Catalog Mode with the console in non-cursor operation the cursor symbol must be displayed if the Note feature is to be used. Under these conditions the typing position and the cursor symbol do not correspond. A special switch was installed adjacent to the console so that if the user wishes he may turn the cursor symbol off when in the Catalog Mode if he does not wish to use the Note feature.

3.3 Monitor Program

The Monitor Program is divided into the Monitor Initialization sequence and the Monitor Loop as shown in Fig. 2.1 of Chapter 2. The 620i operator starts the Monitor Initialization from the processor front panel. When the Initialization sequence is complete control is passed to the Monitor Loop and all function routines are called from that program loop.

3.3.1 Monitor Initialization

The entry point for the Monitor Initialization sequence is located at STRT (Start) and is simply a jump to SCAT (Start Catalog). SCAT initializes the data-phone interrupt switches for each console and also initializes the Master Interrupt Enable/Disable Switch so that interrupts may occur if data is received from CTSS. Control is transferred to ECAT (Enter Catalog) which fetches the user parameter pointer and calls CATM (Catalog Mode) to place each console in Catalog Mode and then branches to CKC (Check Consoles) to begin the Monitor Loop. CATM causes user parameters MODE and TBIX (Table Index) to be set for Catalog Mode and the Catalog label to be written from core onto the user's dispaly label line. No other user parameters are affected so that all other console status information remains unaltered.

The 620i operator may enter at ECAT if he wishes to place the consoles in Catalog Mode without initializing the CTSS interrupt switches. Or he may enter at CKC if he wishes to restart the Monitor Loop without altering any user parameters and without initializing the interrupt switches. Use of the RESET switch on the 620i front panel disables all interrupt switches.

3.3.2 Monitor Loop

3.3.2.1 Console Service

The monitor loop begins at location CKC (Check Console) with a call to EAK (Enter Any Keyboard). EAK inputs a console request word from the buffer-controller console interface. Each bit of the console request word indicates if a particular user display console has requested service. A priority is assigned to each user and each time through the monitor loop, the console request having the highest priority is serviced. Each user display console has associated with it a user parameter list. When a display console has been selected for service a pointer to the first user parameter for that console is placed in the 620i Index Register (XR). All program references to user parameters are then made relative to the XR. If no console requests are waiting the monitor continues on to its next task. To service a display console request EAK calls KBI (Keyboard Input).

KBI outputs a binary number corresponding to the user console number, to the Buffer-Controller display console interface. The interface issues an interrogate signal to the proper user console which responds with a 40 bit reply to the interface. That 40 bit reply will be input to the 620i as three distinct words under program control. KBI waits for the completion of that data transfer which takes approximately 40 microseconds. The wait is accomplished by use of a sense instruction loop that repeatedly tests a hardware flag for completion of the data transfer to the interface. The first word is input and stored at KBW (Keyboard Word). If the input is zero, an error is assumed and no further action is taken on the request. A sense is made for the second console word and if it is not ready an error is assumed and no further action is taken on the request. The second word is stored in CURW (Cursor Word). If the user console is in the Cursor Mode of operation, PCUR (Process Cursor) is called to extract the Cursor Address from the Cursor Word (CURW). The user display parameters Character Count (CCT), Line Count (LCT), and Drum Sector Word (SWRD) are updated. These updated values will be used for whatever function KBI undertakes to service the user request. If PCUR determines that the Cursor Address is an invalid address, no further action is taken on the display console request. If the user console is not in Cursor Mode, the cursor word is not processed.

Next KBI tests KBW (the first console word received) for a function code. The function code is contained in the eight bits to the right of the sign bit. If there is no function code, KBI processes KBW for a keyboard character. Each console display request should cause either a function request or an ASCII character input, but not both in the same request.

The seven bit ASCII character, right justified in the KBW, is extracted and stored in CHAR (Character). The character is then checked to determine if it is one of the system control characters having special functions in the Intrex environment. The correction characters @, #, and back-space cause KBI to take particular actions. The @ character causes a line of null characters (blank spaces on the CRT display) to be written over the user's current display line, and his user display parameters modified so that his next typing character position will be reset to the beginning of the erased line. Also a 620i core record of characters typed on the current display line are zeroed by a call to FLKS (Flush Keyboard String).

The # and the back-space cause a call to WDCH (Write Drum Character) which writes a null character on the user's display track to remove the last character typed. The user display parameters are modified so that the next typing position will be at the place where the character was just erased. The erased character is also removed from the stored keyboard string in the 620i. If the character received in KBW is not one of the special ones mentioned above, it is written on the user's display page by a call to WDCH (Write Drum Character). WDCH writes a single character on the drum track, line, and character position determined by the user's display parameters.

WDCH also checks to see if the last character typed has caused an End of Page (EOP) condition to occur by attempting to advance the next typing position onto the user's label line. This is accomplished by a call to SNL (Start New Line). If the EOP condition exists SNL sets the the user parameter EOPF (End of Page Flag) then calls CB (Check Buffer) to check user parameter BFUF (Buffer Full Flag) and determine if a CTSS buffer overflow message is waiting to be written on the user label line. CB calls DSPM (Display Message) to generate the message and write it on the proper track, and line.

SNL also calls SEOP (Store End of Page) which causes the End of Page display console status light to be lit if EOPF is set. SEOP calls WSC (Write Single Character) to place the proper ASCII code (021) in character position 56 of drum line 30 (label line) of the user's current display page.

If the last typed character was written in a display character position to the left of position 55 (the last visible character position on a display line), then WDCH updates user parameters CCT and SWRD to point to the display position for the next typed character. If the last typed character was written on character position 55 or to the right of that position and if the user is in Catalog mode, WDCH returns without updating CCT or SWRD. As a result the next typed character will be written in the same position as the last one unless the user takes some evasive action. For modes other than Catalog SNL (Start New Line) is called to advance the line count (LCT) to the next line. It is sometimes convenient, especially when doing system checking, to hold down a key at the console to cause repeated typing which can then continue for the full length of the page with characters overflowing from one line onto the next. In Catalog mode, however, such action causes difficulty because CTSS does not take into account the advancement of the line count due to overflow. Also due to the short display line of 56 characters (character positions 0 through 55) catalog users may wish to type long commands without using the Intrex convention for continuation of the command on the next line. In such a case the overflow characters replace the last visible character on the user's display line until he ends the command with a carriage return. The long command will be sent to CTSS and properly processed even though the excess characters will not remain on the display screen.

Next KBI tests the user mode of operation. If the Mode is Catalog, the input character (CHAR) is sent to CTSS by a call to SNDA (Send A). A CR (015 octal) is converted to a LF (012) before sending because CTSS expects the LF code to mark the end of a typed line; however, ASCII keyboards normally generate the CR code when the Return Key is struck. Intrex instructional materials direct the user to employ the Return Key to terminate a command line.

If the user mode is Text Access and CHAR is not a LF or CR, then the character is placed in the 620i store of keyboard characters using KBS (Keyboard String) which points to the user's unpacked, one character per word, store of characters typed on the current display line. This store is used in Text Mode to supply the required microfiche access information, and in Source Handling Mode to supply function parameters.

If the Mode is Text Access and the Character is a LF or CR, KBI calls GTP (Get Text Parameters) and FPR (First Page Routine) to cause a text request to be queued for the text access system. Afterwards the keyboard string is zeroed by a call to FLKS (Flush Keyboard String). Note that characters typed while in Text Mode are not sent to CTSS.

GTP (Get Text Parameters) extracts from the keyboard string at KBS the user parameters Fiche number (FICH), starting frame Row (ROW) and Column (COL), Number of Pages (NPS) in the document, and Last Frame (LFR), Last Row (LR) and Last Column (LC), for the last page of the document.

FPR (First Page Routine) sets up user parameters for the first page of the microfiche document and calls PDQ (Put Display Queue) to generate and queue the text display command sequence. PDQ calls FTR (Flag Text Request) to check TRQF (Text Request Flag). If TRQF is zero or if TRQF contains the console parameter pointer of the console making the current request, the parameter pointer is placed in TRQF and DRQ (Display Request) is set to zero. That means this display request will be the next one sent to the Text Access Central Station when that device is free. If TRQF was not zero in the above case, then the user's current request replaces his previous request which was never completed. If a user causes a new display request before his last one has been sent to the Text Central Station, it is assumed that he no longer wishes the previous request. He cannot view two microfiche images simultaneously. If the request has already been sent to the Text Central Station but the image sweep has not been completed, then a second display request can enter the user's display queue.

TRQF is a system parameter which points to the user display request to be transmitted next to the Text Central Station. If TRQF is not zero and does not point to the user making the current request, then DRQ (Display Request) for the user making the current request is set by FTR. After FTR has set either TRQF or a user's DRQ, then control is returned through PDQ and FPR to KBI.

As described above, KBI tests the Keyboard Word (KBW) received from the user console for a function code. If KBW contains such a code, then KBI extracts that code and calls FSDE (Function Switch Decode) to decode the request and execute the corresponding function routine. There is a code table for each Mode which contains all the eight bit function codes corresponding to the functions which can be called from that Mode. The codes are packed 2 per 16 bit 620i word. FSDE searches the code table corresponding to the user Mode for a match with the input code. If no match occurs, the input is ignored. The end of the code table is marked by a zero word. Entries in a second table, the function routine address table, for a given Mode correspond by position to entries in the code table. Thus the first entry in the code table for a given Mode corresponds to the first entry in the address table for that Mode. Functions can be disabled or removed by simply placing a zero in the corresponding address table without the necessity of reordering the code table. Control is returned from the function routines to KBI and then to EAK.

When control is returned from KBI and EAK to the monitor loop, a call to TSM (Text Mode) is made for each console to process characters received from CTSS and placed into the user's character buffers by the CTSS interrupt service routines. The action of TSM is described below.

3.3.2.2 Processing CTSS Messages

The monitor loop calls TSM (Text Mode) for each console. If the Mode is Catalog, GFP (Get From Pool) is called to determine if any characters received from CTSS have not been transferred from the user's character buffer to the user's line buffer. If characters remain in the character buffer, TSTY (Time Share Type) is called to process the CTSS message. Each pass through the monitor loop processes a single

message character for each user having unused characters in his character buffer.

Messages from CTSS have the following format. A message begins with a Start Text (STX) character with an octal code of 002. The text of the message in ASCII character code follows the STX. The message is terminated with an End Text (ETX) character coded 003 octal. Inputs to the 620i from CTSS dataphone character buffers must be preceeded by the STX code, otherwise they are discarded as noise inputs by the Buffer-Controller software.

Each call to TSTY processes the next item in the character buffer to determine if it is a CTSS Start Text character. If it is a STX character, OFIN is set to indicate that a CTSS message is in progress. If the character is not a STX, OFIN is checked to see if a message is already in progress and if not, the character buffer output pointer is advanced without transferring the character and control is returned to the monitor loop. Thus input from the dataphone not in message format is dropped. If a message is already in progress, the character is checked for an ETX character. An ETX causes the output pointer to be advanced and OFIN to be reset to zero indicating that no message is in progress. If the character is not an ETX, TMZ (Text for Mode Zero) is called.

TMZ tests for MODE=0 which indicates a remote login sequence and requires a call to POT (Print on Teletype) to print the character on the ASR-33B. If the MODE≠0, MVC (Move Character) is called to move the CTSS character to the user CTSS line buffer. MVC uses OLCP (Output Line Character Pointer) to pack the character into the line buffer, two characters per 620i word. A call to EOLC (End of Line Check) is made to detect an end of line condition. If the line is not ended the CTSS character is checked for the CTSS End of Page character, the Delete (0177 octal). If the Delete is detected SEOP (Store End of Page) is called to flag the EOP condition and to light the EOP status light with a call to WEP (Write End of Page). WEP starts a sequence of nested calls to WSC (Write Single Character), CKSW (Check Sector Word), and WFD (Wait for Drum).

If an end of line was detected, WL (Write Line) is called to write the CTSS message line on the user's current display track. Then, if the character buffer is empty, FMBU (Flush Message Buffer) is called to reset character buffer pointers PTCA and CADR, and to zero the first location of the character buffer. MVC then checks the line buffer contents for a command generated by CTSS for the Buffer-Controller. In the current system only the text display command is generated and processed. Other computer-to-computer commands could make use of the text display command format described in the next paragraph.

The CTSS request for text display is a two line message. Both lines begin with the ESC (Escape) character and end with a LF. On the first line the ESC (octal 033) is followed by the control character DC4 (Device Control 4), octal 024, and a string of ASCII characters giving the text command parameters. The parameter string contains a four digit fiche number followed by a single alphabetic character for the starting frame row number, a two digit starting frame column number and a two digit number indicating the number of pages in the document. The second line begins with an ESC character, followed by an alphabetic "D" and a LF. Thus the ESC followed by another character conveys to the monitor a CTSS generated command. The first line is a command to move the parameter string from the user line buffer into the Keyboard String Buffer. The second line is a command to queue a text display request using the parameters stored in the Keyboard String Buffer.

To process a CTSS text display command, MVC checks the first character in the line buffer for the ESC character used to mark command sequences from CTSS. If an ESC is found in the first position of the line buffer, MVC then checks the second position for either the DC4 character or an alphabetic 'D'. If DC4 is detected, MTAP (Move Text Access Parameter) is called to unpack and move the fiche access number contained in the line buffer into the Keyboard String Buffer. If a 'D' is found, TSDI (Text Display Indicator) is set to indicate that a CTSS text display command has been received.

If an end of line was detected the line buffer contents were written onto the user display page and MVC calls WFD (Wait for Drum) to delay until the drum line write is complete. Then ZLB (Zero Line Buffer) is called to reset OLCP, call SNL (Start New Line) to advance

user's line count, and BLNB (Blank Line Buffer) to clear the user's line buffer. Then MVC returns control through TMZ, TSTY, and GFP to TSM. At this time TSM calls TFAT (Test for Automatic Text) to determine if TDSI has been set indicating that a text display request has been initiated by CTSS. If TDSI is set, ETR (Enter Text Routine) is called to process the text data waiting in the Keyboard String Buffer.

ETR calls GTP (Get Text Parameter) to extract from the Keyboard String Buffer the values of text parameters, FICH, ROW, COL, FFRM, NPS and to set CRPG=1. Next ETR calls FPR (First Page Routine) to place the text request into the user's display request queue at DQ. Then FLKS (Flush Keyboard String) is called to clear the Keyboard String Buffer and to reset the string pointer KBS. Control is returned through ETR, TFAT and TSM to the monitor loop which branches to TRT (Text Routine) to process text transactions with the Text Access Central Station.

The action of GTP and FPR are described in more detail in the preceeding section on Console Service. GTP and FPR may be called by KBI earlier in the monitor loop when the user console is being serviced.

3.3.2.3 Text Access Transactions

Text requests may be initiated by a BRISC user pressing function switches on the Text Display Station when operating in either Catalog or Text Mode, or by typing the fiche access number followed by a carriage return when operating in Text Mode. Alternately text requests may be triggered indirectly by the use of the output text command when operating in Catalog Mode. In this case CTSS generates a text display request which is sent to the 620i for processing.

Use of the text function switches is detected by the Monitor when servicing the BRISC terminals. The keyboard input routine KBI extracts the function code and the decoding routine FSDE calls the function routine which generates the display commands from the user's current text parameters and places the commands in either the text display or text copy queue.

Display requests initiated by typing the document access number while operating in the Text Mode are detected by KBI when the carriage return terminates the typed line. The text parameters are

extracted from the core record of the typed keyboard string and the display commands are generated and placed in the display queue.

Text requests generated by CTSS are detected by the routine MVC when CTSS messages are being processed by the monitor loop. While the CTSS message containing the fiche access number is being written on the display page, a copy of the access number string is moved from the user's line buffer into the keyboard string area and a text display request indicator is set to inform the Monitor that a display request has been initiated. Later in the monitor sequence the text request is noted and the text parameters are extracted from the keyboard string buffer and used to generate text commands which are placed in the text display queue.

The actual transactions with the text access system take place later in the monitor sequence when the text command words are taken from the queues and passed to the text access interface for transmission to the Text Access Central Station.

Each user has a single text display request buffer. All of the user display request buffers taken together constitute the text display queue. When a text display request is next to be processed the Text Request Flag (TRQF) is set by the Monitor to a value which indicates the console making the request. A fixed priority determines which of the consoles waiting for display requests is taken first.

When the monitor program determines that the Text Access System is free the Text Request Flag (TRQF) is consulted and the text request from the indicated user text request buffer is sent to the Text Access Central Station so that the fiche may be retrieved, the page positioned and scanned, and the image transmitted to the Text Display storage tube. The monitor program outputs the three word text command sequence to the text interface buffer, sets the Text Access Request Pending Flag (TARP) and then continues on to other duties.

The Text Access interface hardware sends a request for service to the Text Access Central Station. When the Central Station is ready it replies with a Handshake code message to the Text Reply Buffer. The hardware interface transmits the three word text command sequence, which has been waiting in the text interface command buffer, when the handshake has been received by the interface. The identification of

the storage tube terminal to which the image is to be sent is contained in the three word request.

When the monitor program next checks the Text Reply Buffer the reply is decoded by the program and the handshake message is displayed for the user on the label line of the refreshed CRT of the BRISC terminal which made the request. And again the monitor can tend to other work.

After the fiche has been retrieved and the image sent to the Text Display storage tube, the Text Central Station then sends a Ready message code to the Text Reply Buffer. If the image was not sent, an error message code is sent to the Text Reply Buffer. The message is decoded and a corresponding message is displayed for the user on the refreshed CRT label line. Under normal conditions the Monitor proceeds rapidly enough so that the Handshake code is removed from the Text Reply Buffer long before the end of operation message arrives.

It should be noted that text page images from the microfiche document are written and stored on the Text Display Station storage tube screen. Text Reply message codes are decoded and a corresponding message is displayed on the label line of the refreshed CRT of the Catalog Display Station. Thus both of the BRISC CRT display screens are used in the transaction. The video page images sent directly to the storage tube do not affect the 620i monitor program. The text coded message replies do reach the monitor program for processing and they also affect the condition of the lighted buttons on the Text Display Panel. Error codes light the yellow error indicators such as FICHE NOT FOUND and others. These lights are lit by hardware sensing of the coded error messages, not by the monitor program.

While the Text Access Central Station is controlling the retrieval of the fiche for one request the monitor program selects the next request in the display queue to be processed by sett ing the Text Request Flag (FRQF). Text Requests are taken from the display request queue in a fixed order; however, the next request is flagged before the previous request is complete so that consoles with a lower priority can be serviced even if a higher priority console makes new requests as soon as the last request is complete. This scheme is adequate for the current system.

If there are not text display requests in the text display queue then the Monitor processes the text film requests. The film request queue is a first-in, first-out queue with a current capacity for a total of ten film requests. Unlike display requests, it is possible for users to stack up film requests. Since all film requests go into the same queue a single user could fill the queue and so prevent others from queuing a request until one of the film requests is processed. In the current system film requests are not often made and the queue is adequate.

When the current pass at text access transactions is complete a branch to CKC is made to restart the monitor loop program sequence.

3.3.3 CTSS Interrupt Service

An input received from CTSS via th. dataphone interface generates an interrupt request to the 620i processor. If the Master Interrupt Enable Switch and the device interrupt enable for the inputting dataphone are both set, then the interrupt occurs and the processor takes its next instruction from the interrupt location (called the interrupt vector) set aside for use by the interrupting device. Each user dataphone has its own interrupt vector. In the current system only two interrupt vectors are instrumented. They are DPI1 and DPI2 (Dataphone Input 1 and 2) corresponding to inputs to the 620i from dataphones 1 and 2 for BRISC consoles 1 and 2.

If the content of the first locations in the interrupt vector is a Call (JMPM instruction), then the 620i places the contents of the program counter (PC) into the Call address so that the interrupted program may be resumed by the normal indirect return from the called service routine. The execution of the JMPM instruction also disables the Master Interrupt Enable Switch so that no other interrupt can occur

until the Master Enable is re-enabled. The interrupt vectors DPI1 and DPI2 contain a Call to ETS1 and ETS2 (Enter Time Share 1 and 2) respectively. When an input from CTSS causes an interrupt to occur, the contents of the PC are placed at ETS1 or ETS2. Program execution can be resumed by an indirect return through ETS1 or ETS2, however, the Monitor software always uses ETS1 as a return address. If the interrupt is by Dataphone 2 and the resulting Call is to ETS2, the dataphone interrupt service routine moves the return address from ETS2 to ETS1.

The contents of the A, B and X registers at the time of the interrupt are saved by the service routine so that they may be restored when the interrupted program is resumed.

A call to ETS1 results in the fetching of the user parameter pointer for console number 1 and a call to ETS2 in that for number 2. Employing the user parameter pointer the interrupt service routine obtains the correct dataphone input instruction from user parameter TSS1 (Time Share System 1) and inputs the contents of the dataphone interface input character buffer. If the input character is zero, it is ignored. Non-zero inputs are stored, one character per word, in the user CTSS character buffer by use of CADR (Character Address). Then LCAD (Last Character Address) is used to determine if the buffer is full and if so BFUF (Buffer Full Flag) is set. Otherwise CADR is incremented and the next buffer location is cleared. Then the contents of the A, B and X registers are restored to their values prior to the interrupt. The Master Interrupt Enable Switch is set and the interrupted program is resumed by ,RETU*, ETS1.

3.4 Remote Login

The Remote Login Sequence can be entered from either EC1 or EC2 (Enter Console 1 or 2). The sequence allows the system programmer/operator to communicate with CTSS from the Teletype ASR-33B at the 620i site. The entry point used associates with the teletype the user parameters of the corresponding BRISC terminal and dedicates the sequence to that console's dataphone. In particular the user's CTSS character buffer is employed to buffer the CTSS messages arriving at the rate of 1200 baud (or 120 characters per second). Unlike the normal

Monitor, this Teletype (TTY) monitor cannot output characters fast enough to prevent a buffer overflow if CTSS sends messages exceeding the user character buffer length. Hence with the current buffer size the Teletype cannot be used as an Intrex search terminal unless either the buffer length is changed or the CTSS page length is shortened when the login is performed. The loop is designed for logging in remote BRISC terminals from the 620i site and for system checking procedures.

After the entry at EC1 or EC2 the corresponding user parameter pointer is obtained and the dataphone address for the sense Line Free instructions is set for the console number indicated by the entry point. The console number is also stored at GC+1 (Get Character) as part of the calling sequence for GFP (Get From Pool).

The user parameter MODE is set to zero to indicate the Teletype Login Mode and the teletype monitor loop begins at TSTL (Test Line). If the dataphone line is busy, then the program executes a wait loop until CTSS frees the data line. The CTSS input characters generate interrupts which are serviced by the interrupt routine that places the characters in the user CTSS character buffer.

When the data line becomes free the condition of the 620i Sense Switch #1 (SSW1) is checked. If the switch is ON (in the UP position on the 620i panel), then control is passed to GC (Get Character) to process the CTSS message. If SSW1 is OFF, the wait loop at TSTL is entered. The switch is provided to allow the system operator to prevent printing of the message on the TTY until the transmission is complete. This is useful in system checkout and in trouble shooting equipment malfunctions.

The call to GFP (Get From Pool) results in a single CTSS character being output to the TTY for printing. The operation of GFP is described in the paragraphs about the Monitor Loop processing of CTSS messages. Since MODE=0, POT (Print on Teletype) is called to output to the TTY instead of the BRISC.

Next the dataphone line is again tested and if it is busy the wait loop is entered at TSTL+2. Otherwise the TTY input buffer is tested at KBE (Keyboard Enter) to determine if any character has been entered from the TTY keyboard. If not, control is returned to the wait loop at TSTL. If a character is waiting in the TTY buffer it is input

to the 620i and echoed back to the TTY for printing. Then the character is checked for a carriage return (CR) whose presence generates a call to OCRL (Output CR and LF) and a LF (Line Feed) character is substituted to be sent to CTSS by SNDA (Send A). Characters other than CR are sent unaltered to CTSS. Then control is returned to GC to process any CTSS characters waiting in the character buffer.

3.5 Remote Initialize

SBAR (Start Barker) is the starting location of a program sequence for initializing the BRISC terminal in Barker Library after the Remote Login is complete. The user parameter pointer for BRISC 2 is loaded into the XR and SFP (Select First Page) is called to set the console display on user display page 1. CERS (Call Erase) is called to clear user page 1 and WFD (Wait For Drum) to wait until the Drum Unit is free. The sequence is ended by a branch to STRT which is the entry point for the Monitor Initialization sequence. To complete the remote operation the remote ATTN/QUIT button on the TTY may be pressed and the resulting message will appear at the top of page 1 on the Barker BRISC display. The results may be observed at BRISC 1 by placing the console in the Slave View Mode.

IV PARAMETER AND SUBROUTINE DESCRIPTIONS

4.1 User Parameter Descriptions

Each user at a BRISC terminal is associated with a set of software parameters corresponding to the operating status of the console. Their values constitute the most recent history of the user's actions and of the system responses to that user. There is a list in core of such parameters for each user. An index of pointers to the user parameter lists is located at PL (Pointer Lists). PL+1 points to the first entry in the list of User No. 1 parameters; PL+2 to the first entry for User No. 2. The most frequently used mechanism to fetch a user parameter for analysis is shown below:

```
,LDB, L2              Place User No. in B Register
,LDX, PL, BR          Fetch User Parameter Pointer
                      into X Register
,LDA, MODE, XR        Load User Parameter MODE into
                      A Register
```

In the example a program constant L2 = 2 is loaded into the B Register. Next the pointer list at PL is accessed, indexed through the B Register. Thus the contents of PL+2 is placed into the X Register. PL is in lower core below location 512 (decimal). Next the user parameter at MODE is loaded into the A Register for processing. MODE is an offset to the Base taken from PL+2 and is defined in the assembly program with an EQU. The offset must be a positive number less than 512 (decimal).

```
MODE,EQU, 7
```

Thus the base address, or pointer, is held in the X Register while the offset or relative address is assembled into the LDA instruction.

The following paragraphs enumerate the user parameters and describe their functions. The parameter descriptions are grouped according to function as follows: Display Page Status, Buffer Addresses, Mode Status, Saved Parameters, CTSS Transaction Controls, Edit Pointers, Text Access Status.

4.1.1 Display Page Status

4.1.1.1 CCT Character Count:

Points to the next character position to be used on the user's current display page. The left-most character position on a display line corresponds to CCT = 0. There are 64 (decimal) character positions per line; however, only 56 correspond to positions visible on the CRT screen.

4.1.1.2 CUP Current Page:

Contains two ASCII characters giving the users current page number. If the page number is less than 10 (decimal) the left side contains a Null (000 octal) character. If the page number exceeds 9 (decimal) the left side contains an ASCII 1 (061 octal).

4.1.1.3 CURF Cursor Flag:

A two valued indicator which when set to 1 indicates that the terminal is operating in Cursor Mode and when BRISC keyboard input is taken into the 620i the Cursor Word will be processed to yield a new line number (LCT) and character position (CCT) to be used for the current operation. If CURF = 0 then the Cursor Word is not processed and the current value of LCT and CCT are used for the operation.

4.1.1.4 EOPF End of Page Flag:

A two valued flag which indicates that the End of Page Condition exists for the terminal if EOPF = 1. Using the Erase or New Page function switch will set EOPF = 0.

4.1.1.5 FLTP First Light Position:

The core address of the location in the second buffer at SADR that corresponds to the first invisible character position on a refreshed CRT display line. When writing a label line this position is used for the first of a string of characters to light the user's console status lights. The address refers to character position 56 (decimal-starting with the first position numbered zero) or to the left side of word 28 (decimal-starting with word zero).

4.1.1.6 FTRK First Track:

A drum track command word which corresponds to the users first display page. FTRK = 060000 (octal) corresponds to

user 1's first page which normally is Track 96 (decimal). This parameter may be manually changed to a larger number to reduce the number of tracks available to a given user. (See Appendix B for track command word format.)

4.1.1.7 LBSW Label Sector Word:

The drum sector command word used to write the user's current label line from his second line buffer.

4.1.1.8 LCT Line Count:

Points to the current display line to be used in the next BRISC operation when not in Cursor Mode. In Cursor Mode the Cursor Word is processed to obtain a new LCT to be used in the operation in progress. LCT = 0 corresponds to the top display line. LCT = 30 (decimal) corresponds to the last display line above the label line.

4.1.1.9 LTRK Last Track:

A drum track command word which corresponds to the users last dialog display page. Normally this is his Page 14. For user 1 LTRK = 066400 (octal) which corresponds to Track 109 (decimal).

4.1.1.10 NLBF Need Label Flag:

A two valued flag which indicates when a message has been written on the display label line to inform the user of some system condition. When NLBF = 1 the label line has received such a message and the next BRISC keyboard action will re-establish the normal label line for the current operating mode. When the normal label has been restored NLBF is set to zero.

4.1.1.11 SMTK Sample Track:

A drum track command word which corresponds to the user's Sample Display Page. For user 1 SMTK = 067000 (octal) which corresponds to Track 110 (decimal).

4.1.1.12 SW1 Sector Word 1:

Drum sector command word to write Line 0 from users second line buffer (with pointer at SADR). If the contents of SADR were changed then the contents of LBSW, SW1, SW2 and SW3 would all have to be changed. Used in the routine CERS.

4.1.1.13 SW2 Sector Word 2:

Drum sector command word to write Line 1 from users second line buffer (with pointer at SADR). Used in the routine CERS.

4.1.1.14 SW3 Sector Word 3:

Drum sector command word to write Line 31 (the display synch line) from users second line buffer (with pointer at SADR). Used in the routine ERAS.

4.1.1.15 SWRD Sector Word:

A drum sector command word which corresponds to the user's current display line number (LCT) and character position (CCT). This parameter is altered as the user's typing position on the page is altered by user or system actions. (See Appendix B for sector command word format.)

4.1.1.16 TWRD Track Word:

A drum track command word which corresponds to the user's current display page. This parameter is altered as the user selects display pages by BRISC terminal operations such as Next Page, New Page, etc. (See Appendix B for track command word format.)

4.1.2 Buffer Addresses

4.1.2.1 KBS Keyboard String:

Points to next location to be used to store next character received from user keyboard. Characters are stored unpacked, one character per 16-bit word right justified.

4.1.2.2 LKBS Last Keyboard String:

Points to last location available for use in the Keyboard String buffer.

4.1.2.3 OLCP Output Line Character Pointer:

Points to location in buffer to receive next given character. Used in MVC (Move Character) to pack characters (two characters per 16-bit word).

4.1.2.4 PADR Permanent Address:

Starting Address of First Line buffer which is used to write messages received from CTSS onto the users current display page. Used to initialize buffer pointers.

4.1.2.5 PKBS Permanent Keyboard String:

Starting Address of Keyboard String. Used to initialize KBS, the pointer for the unpacked store of characters typed from user's console keyboard.

4.1.2.6 SADR Second Address:

Starting Address of the Second Line buffer. Used for writing the label line on user's display track. Also used as line buffer by ERAS (Erase) and EL (Erase Line).

4.1.3 Mode Status

4.1.3.1 MODE Mode:

Indicates the mode of operation of a given user console. The value of MODE determines the way keyboard inputs are handled by the monitor program. MODE=1 corresponds to Catalog Mode. For all modes other than Catalog the value of MODE is the drum line number of the label line prototype for the given mode. The label line prototype is stored on LBTK (Label Track) which is a system track.

4.1.3.2 TBIX Table Index:

Points to an entry in an index of pointers to function code tables. The value of TBIX indicates the STATUS of the console which determines how function codes are decoded. TBIX is an offset to be added to the base LFNT (Location Function Table) to reference the starting address of a function code table corresponding to a given console status. When TBIX is added to LAT (Location Address Table) a reference to the starting address of an Entry Point Table is generated.

4.1.4 Saved Parameters

The following parameters are used to save the values of specified user display parameters when the user makes a transition from Catalog to View or Edit Mode. The return to Catalog mode then restores the saved values. Provision was made for only one set of saved parameters.

4.1.4.1 VCCT Saved Character Count:

Saves value of CCT.

4.1.4.2 VCUF Saved Cursor Flag:

Saves value of CURF.

4.1.4.3 VEOP Saved End of Page Flag:

Saves value of EOPF.

4.1.4.4 VLCT Saved Line Count:

Saves value of LCT.

4.1.4.5 VMOD Saved Mode:

Saves value of MODE.

4.1.4.6 VSWD Saved Sector Word:

Saves value of SWRD.

4.1.4.7 VTBX Saved Table Index:

Saves value of TBIX.

4.1.4.8 VTWD Saved Track Word:

Saves value of TWRD.

4.1.5 CTSS Transaction Controls

4.1.5.1 BFUF Buffer Full Flag:

Indicates that the user's CTSS character buffer is full. BFUF=1 indicates that there is no room left in the buffer and flags the monitor to write a warning message on the user's label line.

4.1.5.2 CADR Character Address:

Pointer to the next location in the user's character pool to be used for storage of the next message character received from CTSS. CADR is the character input pointer.

4.1.5.3 LARG Large:

Records the largest value that CADR has reached. Used to check adequacy of CTSS message buffering.

4.1.5.4 LCAD Last Character Address:

The address of the last location in the user's CTSS character pool. Used to prevent buffer overflow into program areas of core.

4.1.5.5 OFIN Overflow Indicator:

Indicator used in processing CTSS messages. OFIN=1 indicates that a CTSS message is in progress. OFIN is set to 1 by detection of the STX (002) character and is set to 0 by detection of the ETX (003) in the input stream.

4.1.5.6 PCAD Permanent Character Address:

Starting Address of user's CTSS character buffer. Used to initialize CADR and PTCA.

4.1.5.7 PTCA Print Character Address:

Pointer to next character to be moved from CTSS character buffer into line buffer in preparation for drum line write and display on users current display page. (Print character is a name left from early system in which characters taken from character pool were then printed on the teletype.) PTCA is the output pointer of the CTSS character buffer.

4.1.5.8 SDP1 Send Data Phone 1:

Sense instruction to determine if CTSS has released the data-phone line.

5.1.5.9 SDP2 Send Data Phone 2:

Output instruction to send user typed character to CTSS data-phone interface for transmission to CTSS via phone-line.

4.1.5.10 SDP3 Send Data Phone 3:

Sense instruction to determine if data phone interface send buffer is ready to receive a new character for transmission to CTSS.

4.1.5.11 TSS1 Time Share System 1:

Input instruction to input character received from CTSS data-phone.

4.1.6 Edit Pointers

4.1.6.1 SVLB Save Line Bottom:

Pointer to the next line on Save Page to be used by SVL (Save Line). SVLB is the Save Pointer.

4.1.6.2 SVLT Save Line Top:

Pointer to the next line on the Save Page to be used by RTL (Return Line). SVLT is the Return Pointer.

4.1.6.3 SVTK Save Track:

Track command word for the user's Save Page. For user #1 SVTK=067400 (octal) which corresponds to Track 111 (decimal).

4.1.7 Text Access Status

4.1.7.1 COL Column:

The microfiche document frame column number from the last text request. Column is a binary number which has a range of 1 to 12 (decimal) inclusive.

4.1.7.2 CRPG Current Page:

The microfiche document page number of the last text request. The starting frame of the document is taken as page 1.

4.1.7.3 DQ Display Queue:

The first word of a three-word text access display command sequence corresponding to the last display request made at the user console. The three words remain unaltered until a new display request is made from the same user console. When the three words are placed in the Display Queue, the Display Request Flag (DRQ) for that console is set.

4.1.7.4 DRQ Display Request:

Indicates that a three-word text display request command sequence is waiting in the user's Display Queue at DQ to be sent the Text Access Control Station. DRQ is set to 1 when a user makes a display request and DRQ is reset to 0 when the user parameter pointer is placed in TRQF (Text Request Flag) to indicate that the request at the user's DRQ is to be transmitted next to the Text Control Station.

4.1.7.5 FICH Fiche:

The fiche number from the last text request in the format required for a text access central station command word. The fiche number format is a three-digit (four bits per digit) modified BCD number right justified in the 16-bit word. The high order digit can take on values from 0 to 17 (octal), hence is not a true BCD digit.

4.1.7.6 FFRM First Frame:

The microfiche document frame number for the first page of the last text display request from that user console. The low order four bits are the column number and the next three bits are the row number of the first frame of the document.

4.1.7.7 FRME Frame:

The microfiche document frame number for the last display request made from the user's console. The low order four bits are the column number and the next three bits are the row number of the current page of the document.

4.1.7.8 LC Last Column:

The microfiche document frame column number of the last page of the document which was last requested from the user console.

4.1.7.9 LFR Last Frame:

The microfiche document frame number for the last page of the last text request made from the user console. The low order four bits are the column number and the next three bits are the row number of the last frame of the document.

4.1.7.10 LR Last Row:

The microfiche document frame row number of the last page of the last text request made from the user console.

4.1.7.11 NPS Number of Pages:

The number of pages in the microfiche document which was last requested from the user console. The value of NPS is obtained either from the microfiche access number received from CTSS in response to an 'output text' command or from the Keyboard String (KBS) generated by the user when in Text Access Mode.

4.1.7.12 OSEC Octal Sector:

Not used in the current system. Originally used to hold the sector number contained in SECT in the form of a binary number from 1 to 9 inclusive. A holdover from the time Text Access commands were typed in from the BRISC keyboard rather than generated by the Text Station function switches.

4.1.7.13 ROW Row:

The microfiche document frame row number from the last text request. In the fiche access number the starting row is given as an alphabetic character between A and E inclusive. The user parameter row is a binary number between 1 and 5 inclusive, obtained by taking the low order three bits of the ASCII representation of the alphabetic characters A through E.

4.1.7.14 SECT Sector:

Contains a six-bit display page sector number (a three-bit Row and three-bit Col number) right justified. SECT is used in the third text access command word to indicate the page sector to be magnified. The value of SECT is that created by the user's last Magnify request at the Text Display Console.

4.1.7.15 TDAD Text Display Address:

A program constant from which the first word of the three-word text display command sequence is constructed. The address of the user console to which the display image is to be sent is included in the constant.

4.1.7.16 TDSI Text Display Indicator:

Indicates if a Text Access Request has been received from CTSS. TDSI is set to 1 after the ESC (033 octal) and D (0104 octal) sequence have been detected in the CTSS message stream. TDSI reset to 0 when the text display routine ETR is entered.

4.1.7.17 TFAD Text Film Address:

A program constant from which the first word of a three-word text film copy command sequence is constructed. The address of the user console making the request is included in the constant.

4.2 System Parameters

4.2.1 Console Constants and Variables

4.2.1.1 Display Constants

4.2.1.1.1 LBTK Label Track:

Track Drum Command Word for the system drum track which contains mode label lines.

4.2.1.1.2 STL1 Status Light 1:

Character Code (octal 021) used to light status light 1 on the BRISC terminal from the refresh data derived from the drum.

4.2.1.1.3 STL2 Status Light 2:

Character Code (octal 023) used to light status light 2 on the BRISC terminal from the refresh data derived from the drum.

4.2.1.1.4 MTK Message Track:

Track Drum Command Word for the system drum track which contains system messages.

4.2.1.1.5 CLBS Core Label Sector Word:

Contains the Sector Word Drum Command for writing the string of packed ASCII characters for the Catalog Mode label line from core.

4.2.1.1.6 CLTP Core Light Pointer:

Contains the address of the first location used for storage of ASCII character codes for lighting console status lights from the Catalog label line. CLTP = CL + 28 which corresponds to the first invisible character position on the label-line.

4.2.1.1.7 CL Core Label:

The first location of the string of packed ASCII characters for the Catalog Mode label line in core storage.

4.2.1.1.8 PL Pointer List:

The first location of a list of starting addresses of user parameter lists. The PL location is a dummy so that console numbers begin with 1. A code sequence for obtaining the user parameter MODE for console #2 is:

```
,LDB, L2          Load Console # into BR
,LDX, PL, B       Load X with user parameter pointer
,LDA, MODE, X     Load A with parameter MODE of given
                  user.
```

4.2.1.2 Keyboard Routine Variables

4.2.1.2.1 CONS Console:

Holds the binary console number of the BRISC last serviced through routine EAK (Enter Any Keyboard)

4.2.1.2.2 KBW Keyboard Word:

Holds the keyboard word received from the last console serviced through routine KBI (Keyboard Input).

4.2.1.2.3 CHAR Character:

Holds the last 7 bit ASCII character (right justified) extracted from the last Keyboard Word (KBW) processed for a character input. CHAR and KBW may not correspond because CHAR is generated later in KBI (Keyboard Input) and either an invalid KBW or CURW (Cursor Word) or the presence of a FNWR (Function Word) could cause an exit from KBI without generating a new CHAR.

4.2.1.2.4 FNWR Function Word:

Holds the last function code extracted from KBW (Keyboard Word). The code is contained in the eight bits to the right of the sign bit. The remaining eight bits of the word are zero. FNWR contents may not correspond to KBW contents because FNWR is generated later in the KBI (Keyboard Input) routine than KBW and either an invalid KBW or CURW (Cursor Word) can result in an exit from KBI without generating a new FNWR. If KBW contains an input character and no function code, then FNWR will be left with its contents zero.

4.2.1.2.5 CURW Cursor Word:

Holds the second word of the input sequence from the keyboard taken by KBI (Keyboard Input). Contains an 11 bit Cursor Address consisting of a 5 bit line number concatenated with a 6 bit character position, which locates the position of the Cursor Symbol on the BRISC refresh CRT display.

4.2.1.3 Function Decode Variables

4.2.1.3.1 CDW Code Word:

Contains the last eight bit function code (right justified) processed by FSDE (Function Switch Decode). The remaining eight bits of CDW are zero. FSDE compares the contents of CDW with the codes taken from the function code tables to locate a function routine entry point.

4.2.1.3.2 CDTP Code Table Pointer:

Holds the pointer to code table items used by FSDE in decoding function codes. CDTP is incremented by one when a new pair of packed function codes are needed from the code table. When CDTP causes a word of zeros to be fetched from the code table the search for a match of CDW (Code Word) with a table code is terminated.

4.2.1.3.3 RTP Routine Pointer:

A count of the number of unsuccessful comparisons made by FSDE of CDW contents with code table entries. RTP points to the decoded routine entry point relative to the entry table starting address when a match is made.

4.2.1.4 Erase Routine Variables

4.2.1.4.1 LINC Line Increment:

Program constant used in WXL (Write X Lines) to increment the line number contained in the Sector Word Drum Command. LINC = 01000 is used for writing consecutive drum lines. LINC = 02000 is used to write alternate lines as is done in ERAS (Erase) and EROP (Erase Rest of Page).

4.2.1.4.2 TSW1 and TSW2 Temporary Sector Word 1 and 2:

Locations used by CERS (Call Erase) and ERAS (Erase) to hold Sector Word Drum Commands for even and odd numbered lines. User parameters SW1 and SW2 are placed into TSW1 and TSW2 respectively.

4.2.1.4.3 EVCT and ODCT Even and Odd Count:

Locations used by CERS (Call Erase) and ERAS (Erase) to furnish a line count for the writing of even-numbered and odd-numbered lines.

4.2.2 CTSS Parameters

In the current system each BRISC user has a separate dataphone, CTSS character buffer and line buffer. Thus, currently, all CTSS parameters are contained in the user's parameter set.

4.2.3 Text Access System Parameters

4.2.3.1 Text Flags and Variables

4.2.3.1.1 TRQF Text Request Flag:

Contains the user parameter pointer of the console whose text display request is next to be output to the text access interface for transmission to the Text Access Central Station. Contains a zero if no text requests are waiting to be output.

4.2.3.1.2 TARP Text Access Request Pending:

Contains the user parameter pointer of the console whose text request has been output to the text access interface, but has not yet resulted in a coded end of transaction message from the Text Access Central Station. No new text request can be issued to the interface until TARP is set to zero by either the receipt of a coded text access reply other than the Handshake, or by use of the Text Clear programmable function switch on the BRISC.

4.2.3.1.3 TRPL Text Reply:

Holds the last four bit coded Text Reply received from the Text Access Central Station.

4.2.3.2 Text Copy Queue Parameters

4.2.3.2.1 TAQS Text Access Queue Start:

Contains the address of the first location in the text copy queue area. Used to initialize TAQT and TAQB when setting up an empty copy queue.

4.2.3.2.2 TAQE Text Access Queue End:

Contains the address of the last location,plus one, that can be used in the text copy queue area. TQPU (Text Queue Put) compares TAQE and TAQB to determine if there is more room in the copy queue.

4.2.3.2.3 TAQT Text Access Queue Top:

Contains the address of the next three word text copy command sequence to be sent to the Text Access Central Station. Copy requests are taken from the top of the copy queue and added to the bottom.

4.2.3.2.4 TAQB Text Access Queue Bottom:

Contains the address of the next location to be used when a text copy request is to be placed in the copy queue. Copy requests are added to the bottom of the queue and taken from the top.

4.2.3.2.5 TQCT Text Queue Count:

Contains a count of the number of text copy requests contained in the text copy queue.

4.2.3.2.6 TAFQ Text Access Full Queue:

When set to one indicates that the text copy queue is full. Before additional requests can enter the queue the top item must be removed.

4.3 Program Sequence Descriptions

The following paragraphs describe sections of code that start with an entry point and end with a program branch (a 620i JUMP instruction). Some of the entry points can only be reached by operator action at the 620i console. Other entry points are reached by branches from a preceeding set of program sequences. The most important program sequence is the Monitor Initialization with the entry point Start (STRT) and which branches to the Monitor Loop entry point CKC (Check Consoles).

4.3.1 Initialize Monitor Program

4.3.1.1 STRT Start:

Entry point for starting the Monitor Program. Sequences through SCAT (Start Catalog) to initialize the CTSS Interrupt controls and ECAT (Enter Catalog) which places each user in Catalog Mode. The ECAT sequence branches to CKC (Check Consoles) which is the entry point of the Monitor Loop.

Branches to: SCAT

4.3.1.2 SCAT Start Catalog:

Entry point for initializing the CTSS interrupt controls. SCAT first disables the Master Interrupt Enable Switch and then disables each data-phone interrupt enable switch. Next it re-enables each data-phone interrupt enable switch and lastly the Master Interrupt Enable Switch. SCAT then branches to ECAT (Enter Catalog) to continue the Start Sequence.

Branches to: ECAT

4.3.2 Mode Call Entry Points

4.3.2.1 ECAT Enter Catalog:

Entry point for placing user consoles in Catalog Mode. Calls CATM (Catalog Mode) for each user console to establish user parameters for Catalog Mode and to re-select and re-label the users display page. Next ECAT branches to CKC (Check console) to enter the Monitor Loop. By pressing System Reset at the console and then entering at ECAT the operator may place the user consoles in Catalog Mode with the Interrupt Switches disabled. This is a useful technique for hardware-software debugging.

Calls: CATM
Branches to: CKC

4.3.2.2 ELOC Enter Local:

Entry point for placing user consoles in Local Mode. Calls LOCM (Local Mode) for each user console to establish user parameters for Local Mode and to re-select and re-label the users display page. ELOC then branches to CKC (Check Consoles) to enter the Monitor Loop. The Local Mode is for use by the system programmer-operator for check-out and debugging. TBIX (Table Index) is set to zero and the function switches respond as they do in Catalog Mode; however, there is no transmission to CTSS. Messages received from CTSS are placed in the user character buffer but are not written on the user display page.

Calls: LOCM
Branches to: CKC

4.3.3 Monitor Loop

4.3.3.1 CKC Check Consoles:

Entry point for the Monitor Loop. Calls EAK (Enter Any Keyboard) to take input from the highest priority console and process it. Then calls TSM (Test Mode) for each user console. TSM processes the next CTSS Message Character and tests for CTSS text request. Then CKC branches to TRT (Text Request Test) to continue Monitor Loop.

Calls: EAK, TSM
Branches to: TRT

4.3.3.2 TRT Text Request Test:

Starting point for that portion of the monitor loop which handles transactions with the Text Access Central Station. This program sequence processes coded replies from the text system and supplies the user,whose text request is being processed, messages concerning the request status. Both

display and copy requests are taken from the queues and output to the text access interface as it becomes available. This is the last sequence in the monitor loop and control is passed through TEX (Text Exit) to CKC (Check Consoles) to restart the monitor loop.

Calls: ATR, TQTK
Uses: TARP, TRPL, PL, TRQF, (DQ)
Branches to: CKC

4.3.4 Remote Login

4.3.4.1 EC1 Enter Console 1:

Starting location for logging in BRISC #1 from the Teletype ASR-33B terminal at the 620i site.

EC1 and EC2 sequences join and after an initialization of parameters to dedicate the sequence to the given console, enter a special Teletype monitor loop.

Calls: GFP, SNDA, OCRL
Uses: SEI, PL, (MODE)
Branches to: Internal loop at TSTL or GC.

4.3.4.2 EC2 Enter Console 2:

Starting location for logging in BRISC #2 from the Teletype ASR-33B terminal at the 620i site.

EC1 and EC2 sequences join and after an initialization of parameters to dedicate the sequence to the given console, enter a special Teletype monitor loop.

Calls: GFP, SNDA, OCRL
Uses: SEI, PL, (MODE)
Branches to: Internal loop at TSTL or GC.

4.3.4.3 SBAR Start Barker:

Entry Point for a program sequence used to initialize the BRISC 2 in the Barker library after the remote login is complete. The sequence selects and erases user display page 1 and then branches to STRT to begin the monitor initialization sequence. When the initialization sequence is complete and the Monitor Loop is entered, the Barker BRISC terminal has an empty page 1 display and is operating in Catalog Mode ready for an Intrex search.

Calls: SFP, CERS, WFD
Branches to: STRT

4.3.5 Interrupt Vectors

4.3.5.1 Data Phone Interrupt Vectors

4.3.5.1.1 DPI1 Data Phone Input 1:

Entry point taken by the 620i when an interrupt is generated by input from the CTSS data phone number 1. DPI1 contains a Call to ETS1 (Enter Time Share 1).

Calls: ETS1

4.3.5.1.2 DPI2 Data Phone Input 2:

Entry point taken by the 620i when an interrupt is generated by input from the CTSS data phone number 2. DPI2 contains a Call to ETS2 (Enter Time Share 2).

Calls: ETS2

4.3.6 Utility Entry Points

4.3.6.1 Drum Track Read Write Utility

4.3.6.1.1 CRWI Call Read Write Initiate:

Starting location for Drum Track Read/Write Utility. Start with Drum Command Sector Word in AR, Track Word in BR and number of tracks in XR. Calls RWI (Read Write Initiate) and returns with AR and BR incremented beyond last operation and XR = 0. Waits for Drum Free condition and then Halts. The Halt is followed by a branch to STRT (Start) at location 00000.

Calls: RWI
Branches to: STRT (After Program Halt)

4.3.6.2 Octal Dump Utility

4.3.6.2.1 DNPG Dump N Pages:

Starting address for Octal Dump Utility which prints an octal dump of core contents on the TTY. Calls LD (Load Dump) which then calls DUMPO (Dump Octal) which in turn calls OCRL (Output Carriage Return Line Feed), SPA5 (Space 5), ODMP (Octal Dump) and RPTO (Repeat Output).

Calls: LD
Branches to: Halt (Then RUN restarts the octal dump)

4.4 Subroutine Descriptions

The following descriptions of subroutines are grouped according to function. Within a functional group the routines are in alphabetical order on the mnemonic routine name. At the end of each description the parameters and program variables used by the routine are given. User parameters are enclosed in parentheses to distinguish them from other program variables. Numerical constants used are not listed. Also listed are the names of subroutines called by the routine being described.

The functional grouping of the subroutine descriptions follows: Console Display Service Routines, Drum Routines, Line Buffer Routines, Core Transfers, CTSS Message Processing, Label Message Processing, Mode Change and Mode Entry Routines, Edit Mode Routines, Text Access Routines.

Subroutines are entered by use of the pseudo-operation, CALL, NAME which assembles as a Jump and Mark (JMPM) instruction. The return from the subroutine is made with the pseudo-operation, RETU*, NAME which assembles as an indirect Jump (JMP*).

4.4.1 Console Display Service Routines

4.4.1.1 BLKL Blank Line:

Causes the contents of user's first line buffer to be written on user's current display line. Waits for drum to become free before return. (In order to blank line, the line buffer must be cleared prior to call for BLKL.)

Uses: (LCT), (PADR)
Calls: WL

4.4.1.2 CB Check Buffer:

Checks and clears Buffer Full Flag (BFUF) and writes Buffer overflow Message on user display label line then sets Need Label Flag (NLBF).

Uses: (BFUF), (NLBF), LBFM
Calls: DSPM

4.4.1.3 CERS Call Erase:

Clears user parameter EOPF (End of Page Flag) and calls ERAS (Erase) to write null characters on the first 30 lines of the user's current display page. If the console is in Catalog Mode, an ASCII FF (Form Feed) with octal code 014 is sent to CTSS by a call to SNDA (Send A) to indicate that the terminal has begun a new display page. ERAS also establishes the user display parameter values for display at the top of the newly erased page.

The user parameter BFUF (Buffer Full Flag) is checked to determine if a CTSS buffer overflow has occurred and the user is informed of such an overflow by a call to DSPM (Display Message) to present the overflow message on the

label line of the user's current display page. The BFUF is cleared and Need Label Flag (NLBF) is set so that the buffer overflow message may be removed later when the next user keyboard action occurs.

Uses: (NLBF), (EOPF), (SW1), (SW2), (MODE), (BFUF) EVCT, ODCT, TSW1, TSW2 LBFM

Calls: ERAS, SNDA, DSPM

4.4.1.4 EAK Enter Any Keyboard:

Inputs the console request word, decodes the console number with highest priority requesting service, picks up user parameter pointer and calls Keyboard Input (KBI) for that console.

Uses: CONS
Calls: KBI

4.4.1.5 EL Erase Line:

Clears users second line buffer and writes line of zeros on current user display line.

Uses: (LCT), (SADR)
Calls: BSLB, WL

4.4.1.6 EOLC End of Line Check:

Checks character to be printed to detect the end of display line condition.

Enter with character in AR. If end of line condition is caused by an ASCII CR or LF or by a display line overflow of more than 55 characters, a return is made with AR = 0. Otherwise return is made with negative number in AR.

Uses: (CCT), SA
Calls: None

4.4.1.7 ERAS Erase:

Causes users second line buffer to be zeroed and used to zero users current display page and to write an updated display label line. Drum writes are interlaced with even numbered lines written first, label line next, then odd numbered lines, and lastly the drum synch line. Resets user parameters for typing on first position of top of display page. Flushes user keyboard string buffer.

Uses: (TWRD), (SADR), (SW3), TMX, LINC, TSW1, EVCT, TSW2, ODCT, XS

Calls: BSLB, WXL, GLBL, WFD, ZSWR, FLKS

4.4.1.8 FSDE Function Switch Decode:

Decodes given function code and calls indicated function routine. Employs user's Table Indicator (TBIX) to locate code table appropriate for user's status. If the input code does not match any table entry,a return is made with no action taken. Zero marks the end of the code table. The code table contains eight bit codes packed two to a 620 word. A corresponding call address table contains one address per word. If the call address produced is less than 01000 (octal),no action is taken. If it is a negative number, it is decoded as a text access magnify sector code which generates a text magnify request.

Uses: (TBIX), (SECT), CDW, LFNT, CDTP, RTP, LAT
Calls: MGR

4.4.1.9 GLBL Get Label:

Writes Drum Line corresponding to user's Label Line of his current Display page, updating status information. If user is in Catalog Mode,GLBL writes drum line from core buffer CL. For Modes other than Catalog GLBL reads the correct label line from the label track (LBTK) into the user's line buffer at SADR, updates user status,and writes the display label line from SADR.

Uses: (MODE), (SADR), (FLTP), (LBSW), (TWRD), (FTRK), (CUP), LBTK, CLBS, CLTP, DS, DM

Calls: FLSW, WFD, LLL, GSTL

4.4.1.10 GNWP Get New Page:

Advances user's display page number by a call to SNP (Start Next Page). If the user parameter TWRD (Track Word) indicates that the user is not on his last display page as indicated by LTRK (Last Track),a call to CERS (Call Erase) will erase the contents of the next page, reset user parameters in preparation for typing at the top of the display page, and send an ASCII Form Feed (FF) character (014 octal) to CTSS to signal that a new page is ready. If the user is on his last display page, his first display page is selected and GNWP calls DSPM (Display Message) which displays a message on the display label line informing the user that he is on page 1 and may press Erase to continue. GNWP may be called only when the user console is in Catalog Mode.

Uses: (TWRD), (LTRK), LERM
Calls: SNP, DSPM, CERS

4.4.1.11 GSTL Get Status Lights:

Places ASCII Control Codes for current active status lights in line buffer. Used in updating user display label line.

Enter with FLTP Pointer to line buffer store location for first status code.

Uses: (EOPF), (CURF), STL1, STL3
Calls: None

4.4.1.12 KBI Keyboard Input:

Takes input from user display console and processes it. Causes typed character to be written on drum for display on users current page. Processes and uses Cursor Address input if console is in cursor mode operation. Decodes Function Switch Code inputs and calls Function routines. Processes Intrex correction characters # and backspace (delete last character), and @ (delete last line). Stores keyboard characters in keyboard string for catalog and text modes. Outputs keyboard character to CTSS data phone interface in Catalog Mode. Gets Text Parameters from Keyboard String when a CR is encountered while in Text Mode.

Uses: (CURF), (MODE), (CCT), (SWRD), (KBS), KBW, CURW, FNWR, CHAR, XXX, SXX

Calls: PCUR, FSDE, WDCH, SKBS, SNDA, BLNB, FLKS, BLKL, GTP, FPR

4.4.1.13 NTLN Note Line:

Note line calls SVL (Save Line) to transfer a line of the user's current display page to the user's note page (same physical drum track as the user save page). If the display console is not in Cursor Mode when NTLN is entered, NTLN calls VPAR (Save Parameters) and PCUR (Process Cursor) to save user display parameters and then process the Cursor Word to determine the number of the display line to be saved. (The user places the cursor on the line to be saved before pressing the Note Line function switch.) When the Note Line function switch is pressed the cursor word will be processed when the keyboard input occurs in KBI (Keyboard Input) if the console is in cursor mode. If the console is not in Cursor Mode the cursor address will not be processed in KBI so NTLN must make a separate call to PCUR to obtain the cursor address. When this happens RSTP (Restore Parameters) is called to restore the display parameters saved prior to the call to PCUR.

Uses: (CURF)
Calls: SVL, VPAR, PCUR, RSTP

4.4.1.14 PCUR Process Cursor Word:

Extracts character position and line number from the Cursor Word and generates a new drum sector word (SWRD). If line number exceeds 31 returns with AR = 1.

Uses: (CCT), (SWRD), (LCT), CURW, TLCT
Calls: None

4.4.1.15 SEOP Store End of Page:

Checks user parameter End of Page Flag (EOPF) and causes the display console End of Page Status Light to be lit by a single character drum write.

Uses: (EOPF)
Calls: WEP

4.4.1.16 SFP Select First Page:

Selects and labels user's first display page. Updates user track word (TWRD).

Uses: (FTRK), (TWRD)
Calls: TSLB

4.4.1.17 SKBS Store Keyboard String:

Stores unpacked the last display console keyboard input character (CHAR) in the user keyboard string buffer. Clears the next buffer location and increments the keyboard string pointer KBS. If the string buffer is full,the last buffer location is used again.

Uses: (KBS), (LKBS), CHAR
Calls: None

4.4.1.18 SMST Set Mode Select Track

Updates user parameters MODE and TBIX (Table Index) then calls TSLB (Track Select and Label) which selects user display page and updates the display label line.

Uses: (MODE), (TBIX), (TWRD)
Calls: TSLB

4.4.1.19 SNL Start New Line:

Increments user display parameters Line Count (LCT) and Sector Word (SWRD) and sets Character Count (CCT) to zero in order to advance user's typing position to the beginning of the next line. If starting a new line creates an End of Page condition,the user End of Page Flag (EOPF) is set by a single character write,and the typing position is not altered. Checks for CTSS Buffer Overflow Flag and causes overflow message to be displayed on the user display label line.

Uses: (LCT), (EOPF)
Calls: CB, SEOP

4.4.1.20 SNP Select Next Page:

Selects and labels the next consecutive user display page. Wraps around from user last page to user first page. Updates user Track Word (TWRD).

Uses: (TWRD), (LTRK)
Calls: SEP, TSLB

4.4.1.21 SPP Select Previous Page:

Selects and labels the previous user display page. Wraps around from first to last page. Updates user Track Word (TWRD).

Uses: (TWRD), (FTRK), (LTRK)
Calls: TSLB

4.4.1.22 TFML Test for Max Line:

Enter with line number in AR. If line number is less than 29,return with line number in AR otherwise return with AR = 29.

Calls: None

4.4.1.23 TSLB Track Select and Label:

Re-selects user's display page and updates and re-writes users display label line.

Calls: WFD, GLBL

4.4.1.24 TSM Test Mode:

Tests for Catalog Mode of a given user console and processes CTSS messages. Checks for text request from CTSS.

Uses: (MODE), PL
Calls: GFP, TFAT

4.4.1.25 WDCH Write Drum Character:

Writes a single character on the drum track sector corresponding to the users current display track, line count (LCT) and character count (CCT). Checks for and processes end of page and needed label conditions.

Uses: (TWRD), (LCT), (SWRD), (CCT), (MODE), (NLBF)
Calls: CKSW, WFD, SNL, GLBL

4.4.1.26 WEP Write End of Page:

Causes the user display console End of Page status light to be lit by a single character drum write.

Uses: EPSW, STL1
Calls: WSC

4.4.1.27 ZSWR Zero Sector Word:

Sets user's Line Count (LCT), Character Count (CCT), and Sector Word (SWRD) to zero. Used to place typing position at top of display page.

Uses: (CCT), (LCT), (SWRD)
Calls: None

4.4.2 Drum Routines

4.4.2.1 CKSW Check Sector Word:

Compares Drum Sector Counter to a given Line Number. Enter with Line Number in AR. If Drum Sector Counter is within 4 sectors of the beginning of the given line, returns with AR = 0, otherwise AR = 1. CKSW is used to delay Drum selects until desired sector is nearly under the read/write heads in order to reduce CRT Display blanking period encountered during Drum read/write operations.

Uses: VLCT
Calls: WFD

4.4.2.2 FLSW Form Line Sector Word:

Generates Drum Line Sector Word Command for Drum read/write operation. Enter with Core Line Buffer starting address in BR and Drum Line Number in AR. Returns with generated sector word in AR. If Line Number exceeds 31 the sector word is generated for Line 29 corresponding to display line above the Label Line.

Uses: MEM
Calls: None

4.4.2.3 LLL Locate Last Line:

Delays program execution until the drum line corresponding to the label line of the display is within read/write position limits.

Calls: CKSW

4.4.2.4 WD Write Drum:

Waits for Drum not busy condition then issues given DRUM sector and track commands.

Calls: WFD

4.4.2.5 WDL Write Drum Line:

Causes Drum Sector Word to be generated and program delayed until sector counter indicates write position about to pass under write heads, then issues write drum line commands.

Uses: (TWRD)
Calls: FLSW, WFD

4.4.2.6 WFD Wait For Drum:

Delays program execution until Drum Busy condition clears

Calls: None

4.4.2.7 WFL Wait For Line:

Delays program until Drum Sector Counter indicates that the given line is about to pass beneath the drum read/write heads. Enter with given line number in AR. Return with given line number in AR.

Uses: ?LCT
Calls: CKSW

4.4.2.8 WL Write Line:

Delays program until given line is written on the drum.

Uses: SLCT
Calls: CKSW, WDL

4.4.2.9 WSC Write Single Character:

Writes a given single character on the drum track corresponding to the user display page at a given character position. Enter with single character to be written in the AR and the Drum Sector Word Command in the BR.

Uses: (TWRD), STW, SSW, LN
Calls: CKSW, WFD

4.4.2.10 WXL Write X Lines:

Issues a given number of given read/write drum line commands to drum controller. The parameter LINC is used to increment the drum line number. Enter with drum sector word in AR, drum track word in BR and number of tracks in XR.

Uses: LINC
Calls: WFD

4.4.3 Line Buffer Routines

4.4.3.1 BLNB Blank Line Buffer:

Loads zeros into user's first line buffer.

Uses: (PADR)
Calls: RPCR

4.4.3.2 BSLB Blank Second Line Buffer:

Loads zeros into user's second line buffe

Uses: (SADR)
Calls: RPCF

4 4.3.3 FLKS Flush Keyboard String:

Causes user keyboard string buffer to be loaded with zeros and the pointer to string store position (KBS) to be reset to the first position of the string buffer.

Uses: (PKBS), (KBS)
Calls: RPCR

4.4.3.4 ZLB Zero Line Buffer:

Waits until the Drum Busy condition clears, then causes the user first line buffer to be loaded with zeros and sets up new-line conditions.

Uses: (PADR), (OCLP)
Calls: SNL, WFD, BLNB

4.4.4 Core Transfers

4.4.4.1 CTC Core to Core (Same as CTCM):

Moves block of consecutive core location contents. Enter with From Starting Address in XR, To Starting Address in BR, From Ending Address in AR. Returns with Next From Address in XR, Next To Address in BR and AR = 0.

Calls: None

4.4.4.2 RPCR Repeat Core:

Stores given word in consecutive core locations. Enter with word to be stored in AR; the number of locations to be altered in XR; the address of the first core location in BR.

Calls: None

4.4.4.3 UPK Unpack:

Unpacks given packed character string. Enter with input string address in XR, output block address in BR and number of output locations in AR. Assumes first input word contains two characters. Produces a given count of output characters, hence will give an odd or an even number of output characters. Overflow Indicator of 620i used in unpacking. OFL set indicates character in left half of word. UPK saves and restores entering condition of 620i Overflow Indicator by use of temporary storage RO (Restore Overflow).

Uses: NBR, MCT, RO
Calls: None

4.4.5 CTSS Message Processing

4.4.5.1 ETS1 and ETS2 Enter Time Share 1 and 2:

ETS1 and ETS2 are the service routines for interrupts generated by inputs from CTSS to dataphones 1 and 2 which correspond to BRISC consoles 1 and 2. ETS1 and ETS2 are called from the dataphone interrupt vectors at DPI1 and DPI2 (Dataphone Input 1 and 2). If ETS2 is called, the return address at ETS2 is moved to ETS1 and calls to both ETS1 and ETS2 return by execution of a ,RETU*, ETS1. The service routines save the contents of registers A, B and X which are restored before control is returned to the interrupted program. The dataphone input character buffer is read and non-zero inputs are stored in the users unpacked CTSS character buffer in core for later processing. If the buffer is filled, the buffer full flag is set, otherwise the buffer input pointer is advanced and the next buffer location is cleared. The A, B and X registers are restored, the Master Enable is reset and control is returned to the interrupted program with a ,RETU*, ETS1.

Uses: SAI, SAI+1, SAI+2, PL, (TSS1), (CADR), (LCAD), (BFUF)
Calls: None

4.4.5.2 FMBU Flush Message Buffer:

Resets Character Buffer Pointers PTCA and CADR to beginning of Character Buffer and zeros first location in the Buffer.

Uses: (PCAD), (PTCA), (CADR)
Calls: None

4.4.5.3 GFP Get From Pool:

If given user is in either Catalog or Start Mode, that user's CTSS Character Buffer is checked and a call is made to TSTY to process an unused character. If all characters are pro-

cessed, the largest value of the character pointer is saved in LARG before calling FMBU to reset Character Buffer parameters.

Uses: (MODE), (PTCA), (CADR), (PCAD), (LARG), PL
Calls: FMBU, TSTY

4.4.5.4 MVC Move Character:

Packs given CTSS Character into user line buffer using OLCP (Output Line Character Pointer), increments user display Character Count (CCT), tests for Line Feed (LF) and calls WL (Write Line) to write drum line of CTSS message for user display page. If the Character Buffer is empty, MVC flushes message buffer pointers. Checks CTSS line for an ESC (octal 033) character and, if followed by DC-4 (octal 024), calls MTAP (Move Text Access Parameters) to move text parameters received from CTSS from user line buffer (packed) to user keyboard string buffer (unpacked). If the ESC character was followed by an ASCII 'D' (octal 0104), the user parameter Text Display Indicator (TDSI) is set. When the line write is complete the line buffer is cleared.

Uses: (OLCP),(CCT),(PADR),(LCT),(PTCA),(CADR),
(TDSI),(EOPF),SVCH,SXR
Calls: EOLC, SEOP, WEP, WL, WFD, ZLB, MTAP

4.4.5.5 POT Print on Teletype:

Processes given character for teletype (TTY) output. Converts LF (octal 012) to a CR (octal 015), LF sequence by call to OCRL (Output Carriage Return-Line Feed). Maps lower case characters into upper case for TTY. Enter with given character in AR.

Calls: OCRL, OATY

4.4.5.6 SNDA Send A:

Tests for CTSS Data Phone Character Output Buffer to be free then outputs the right most seven bits of the A Register to CTSS.

Uses: SDP3, SDP2
Calls: None

4.4.5.7 TMZ Test for Mode Zero:

If MODE = 0 (Start-up Mode) call POT (Print on TTY) to output CTSS message on teletype (TTY). Otherwise call MVC (Move Character) to process CTSS character.

Uses: (MODE)
Calls: MVC, POT

4.4.5.8 TSTY Time Share Type:

Processes messages in Character Buffer from CTSS. Expects messages to be framed with ASCII Control Characters for Start of Text STX (02) and End of Text ETX (03). Input not preceeded by STX, although taken into Character Buffer, are not printed on display. User parameter OFIN (Overflow Indicator) and 620i Overflow Indicator used to indicate message in process. Calls TMZ (Text Mode Zero) which calls MVC (Move Character) which calls WL (Write Line) to write the line of CTSS message on the drum so that it will appear on the user's display page. Each call to TSTY only works on the next character to be taken from the given user Character Buffer.

Uses: (OFIN), (PTCA), LROF, VX, LIA
Calls: TMZ

4.4.6 Label Message Processing

4.4.6.1 DSPM Display Message:

Obtains message from given line of Message Track and Writes it onto users display page label line with updated status and page information. Sets the Need Label Flag (NLBF). Enter with message line number in AR.

Uses: (CUP), (NLBF)
Calls: GM, WFD, GSTL, WM

4.4.6.2 GM Get Message:

Reads drum line from Message Track into core Buffer. Drum Line number taken from FLNE. Core Buffer Starting Address taken from TADR. Track Word taken from MTK.

Uses: FLNE, TADR, MTK
Calls: CKSW, FLSW, WFD

4.4.6.3 WM Write Message:

Writes message from core buffer onto drum line on user's current page. Enter with drum line number in AR. Core Buffer Sta rting Address taken from TADR.

Uses: (TWRD), TLNE, TADR
Calls: CKSW, FLSW, WFD

4.4.7 Mode Change and Mode Entry Routines

4.4.7.1 CATM Catalog Mode:

Sets AR = 1, BR = 0 and calls SMST (Set Mode Select Track) to update user parameters MODE, TBIX (Table Index) and to reselect the users current display page and update the display label line and console status lights. CATM is the

same as NCAT (New Catalog) except NCAT puts the user console in Non-Cursor Mode while CATM leaves the current Cursor status unchanged.

Calls: SMST

4.4.7.2 CFVE Catalog From View:

Calls RSTP (Restore Parameters) to restore saved user display parameters and then calls NCAT to enter catalog mode. This routine is called when the user returns to catalog mode from the view mode. The display parameters were saved when he went from catalog to view. This is done so that the user may type on the view page, use the cursor, use the editing features available in view mode and then return to catalog mode and have his proper page and typing position, label line and status lights resumed just as he left them.

Calls: RSTP, NCAT

4.4.7.3 CURM Cursor Mode:

Cursor Mode control routine which puts the user console in Cursor Mode if it is currently in Non-Cursor Mode and takes it out of Cursor Mode if currently in Cursor Mode. It either sets or resets the user parameter CURF (Cursor Flag). CURF = 1 corresponds to the console operating in Cursor Mode. CURM calls GLBL (Get Label) to update the user display label line and change the state of the console status light, CURSOR.

Uses: (CURF)
Calls: GLBL

4.4.7.4 EDIT Edit:

Updates MODE and TBIX (Table Index). Selects user display track and updates display label line for Edit Mode.

Calls: SMST

4.4.7.5 FPFV First Page From View:

Calls RSTP (Restore Parameters) to restore saved user parameters then calls SFP (Select First Page) to select and update display label line for the users first display page.

Calls: RSTP, SFP

4.4.7.6 LOCM Local Mode:

Sets AR = 4, BR = 0 and calls SMST (Set Mode Select Track) to update user parameters MODE and TBIX (Table Index) and to reselect the users current display page and update the display label line and console status lights. LOCM is not

used in the current system. The Local Mode of operation was merged with the EDIT Mode to simplify user operation of the console. NLOC replaced LOCM in function entry tables.

Calls: SMST

4.4.7.7 NCAT New Catalog:

Sets AR = 1, BR = 0, clears Cursor Flag (CURF) and calls SMST (Set Mode Select Track) to update user parameters MODE, TBIX (Table Index) and to reselect the users current display page and update the display label line and console status lights. Resetting CURF takes the console out of Cursor Mode when entering Catalog Mode.

Uses: (CURF)
Calls: SMST

4.4.7.8 NLOC New Local:

Calls VPAR (Save Parameters) to save user display parameters. Then sets the Cursor Flag (CURF) and calls the Edit Function routine which places the console in Edit mode. Setting the cursor flag (CURF) causes the console to be placed in Cursor Mode upon entering Edit mode. The name NLOC is a hold-over-from when the corresponding function switch was used to enter a Local Mode which is no longer active.

Uses: (CURF)
Calls: VPAR, EDIT

4.4.7.9 RFV Return from View:

Calls RSTP (Restore Parameters) to restore user display parameters from the save parameter list. Calls SMST (Set Mode and Select Track) to restore the mode from which the View Mode was entered and to generate the corresponding label line and status light conditions.

Uses: (TBIX)
Calls: RSTP, SMST

4.4.7.10 RSTP Restore Parameters:

Restores the values of user display parameters saved in the user save parameter list by a call to VPAR. This is done when returning from certain modes such as returning from Edit to Catalog, or from View to Catalog. Only one level of parameter saving is done. Thus, if the user goes from Catalog to Edit his display parameters are saved on the assumption that he will return to Catalog and so need them. If instead he goes from Catalog to Edit and then to View, the change from Edit to View will save the Edit parameters, destroying the Catalog parameters. Then a return from View to Edit to Catalog will leave him his Edit parameters.

The infrequent occurrance of such action did not appear to warrent establishing a stack of parameter lists with the attendant space and processing time requirements.

Uses: (VLCT), (LCT), (VCCT), (CCT), (VCUF), (CURF), (VEOP), (EOPF), (VTBX), (TBIX), (VTWD), (TWRD), (VSWD), (SWRD), (VMOD), (MODE)

Calls: None

4.4.7.11 VMDE View Mode:

Calls VPAR (Save Parameters) to save user display parameters in the user save parameter list. Sets the Cursor Flag (CURF) to enter cursor mode, sets up the users Save Page for viewing at the display console, sets user parameters MODE and TBIX (Table Index) for the View Mode and calls SMST (Set Mode Select Track) to select the save page and write the View Mode label on the display.

Uses: (CURF), (SVTK), (TWRD), (MODE), (TBIX)
Calls: VPAR, SMST

4.4.7.12 VPAR Save Parameters:

Saves the values of user display parameters in user parameter save locations so that they may be restored later. This is done when the user calls for certain mode changes such as a change from Catalog to Edit, or from Catalog to View. The followed parameters are saved: LCT, CCT, CURF, TBIX, TWRD, SWRD, EOPF, MODE.

Uses: (LCT), (VLCT), (CCT), (VCCT), (CURF), (VCUF), (TBIX), (VTBX), (TWRD), (VTWD), (SWRD), (VSWD), (EOPF), (VEOP), (MODE), (VMOD)

Calls: None

4.4.7.13 VSMP View Sample:

Sets user parameters MODE and TWRD so that a call to SMST (Set Mode Select Track) selects the user Sample Page for viewing at the display and writes the View Sample label line. View Sample mode can only be called from View Mode and since only return functions are available in View Sample Mode no call to VPAR is made to save display parameters. However, the user can type in View Sample Mode and can use the @ for line erase and # for character erase.

Uses: (MODE), (SMTK), (TWRD)
Calls: SMST

4.4.8 Edit Routines

4.4.8.1 CSVP Clear Save Pointers:

Resets the Save Line and Return Line Pointers, Save Line Top (SVLT) and Save Line Bottom (SVLB) to zero.

Uses: (SVLT), (SVLB)
Calls: None

4.4.8.2 EROP Erase Rest of Page:

Clears End of Page Flag (EOPF) and constants Odd Number Line Count (ODCT), Even Number Line Count (EVCT), and Drum Sector Command Words TSW1, TSW2. EROP then calls ERAS (Erase) to store nulls on the user's current display line and on the rest of his display page. Employs user's second line buffer, referenced through user parameters SW1 and SW2 (Sector Words 1 and 2).

Uses: (EOPF), (LCT), (SW1), (SW2), EVCT, ODCT, TSW2, TSW1
Calls: ERAS

4.4.8.3 RTL Return Line:

Reads drum line from users current Save Page. Returns line into users second line buffer then writes that line onto users current display line. Increments Return Line Pointer (SVLT) and clears user's Character Count (CCT), and Sector Word (SWRD).

Uses: (SVLT), (SADR), (SVTK), (LCT), (TWRD), (CCT), (SWRD)
Calls: TFML, WFL, FLSW, WDL

4.4.8.4 SVL Save Line:

Reads drum line from users current display line into user's second line buffer, then writes that line onto current save line of user's save page. Increments Save Line Pointer (SVLB).

Uses: (LCT), (SADR), (TWRD), (SVLB), (SVTK)
Calls: FLSW, WDL, TFML, WFL

4.4.9 Text Access Transaction Routines

4.4.9.1 ATR Any Text Request:

Checks the user text display request flag DRQ for an unprocessed text display request. If DRQ for user #1 is set DRQ is cleared and text parameter TRQF is set with user 1 parameter pointer. If User 1 has no text display request, user #2 is checked.

Uses: (DRQ), PL, TRQF
Calls: None

4.4.9.2 CR Copy Routine:

Calls TQPU (Text Queue Put) to place text film copy request in Copy Queue.

Calls: TQPJ

4.4.9.3 EMG Enter Magnify:

A dummy routine which does nothing. Formerly the Magnify of a micro-fiche page of text was done in several steps, one of which generated the EMG function code. That code is not used in the current system.

4.4.9.4 ETA Enter Text Access:

The text access mode number, 2 is loaded into the AR and the code table number 0 into the BR. A call to SMST (Set Mode Select Track) updates user parameters MODE and TBIX and by a call to TSLB (Track Select and Label) reselects the user's current display page and updates the display label line and console status lights.

Calls: SMST

4.4.9.5 ETR Enter Text Routine:

Calls GTP and FPR to generate and queue text display commands. Flushes user keyboard string where CTSS text request characters were temporarily stored by MTAP.

Uses: X2
Calls: GTP, FPR, FLKS

4.4.9.6 FNB Find Non-Blank:

Locates first character in consecutive locations which is not a blank nor a null. Enter with pointer to first location in XR. TLKB gives address of last location to be tested. Return with AR = 0 indicates non-blank not found. Otherwise, AR contains the first non-blank character at return.

Uses: TLKB
Calls: None

4.4.9.7 FPR First Page Routine:

Sets Users Text Parameters for first page of fiche document. Sets CRPG = 1. Sets FRME, ROW and COL from FFRM. Calls

PDQ (Put Display Queue) to format and queue text display request.

Uses: (CRPG), (FFRM), (FRME), (ROW), (COL)
Calls: PDQ

4.4.9.8 FTR Flag Text Request:

If Text Parameter TRQF (Text Request Flag) indicates that no previous text request is queued,or if the previous request is by the same user, TRQF is set with user's parameter pointer and the user display request flag (DRQ) is cleared. If another user has a display request already flagged in TRQF, then the new user's display request flag (DRQ) is set to indicate that the new request has been placed in the user's display queue.

Uses: TRQF

4.4.9.9 GDEC Get Decimal:

Converts unpacked string of BCD characters into a binary number. Enter with number of digits to be processed in AR.

Uses: ACC, CTD, MD, LCH
Calls: TFE, TDIG, X10

4.4.9.10 GLF Get Last Frame:

Generates Fiche Last Row (LR), Last Column (LC), Last Frame (LFR) from initial values of Row (ROW), Column (COL) and number of Pages (NPS).

Uses: (ROW), (COL), (NPS), (LC), (LR), (LFR)
Calls: None

4.4.9.11 GTP Get Text Parameters:

Extracts from user keyboard string the text parameters FICH, ROW, FFRM, COL, NPS, CRPG.

Uses: (KBS), (PKBS), (FICH), (ROW), (FFRM), (COL), (NPS), (CRPG), TFIC, TROW, TCOL, TNPS, TLKB, SFX, DCT, FXF, LF7
Calls: FNB, TDIG, TFE, DGEC, GLF

4.4.9.12 IDFR Increment-Decrement Frame:

Updates user text parameters in preparation for request for a new text page. Enter with BR = 0 indicates that the fiche frame number is to be decremented as required by the Previous Text Page function. Enter with BR = 1 indicates that the fiche frame is to be incremented as required by the request for the Next Text Page function. If attempts are made to use Next Page when on the last page of the document IDFR calls

DSPM (Display Message) to display the appropriate message on the user display label line and disregards the request. IDFR also updates user text parameters FRME, ROW and COL in accordance with the given request.

Uses: (CRPG), (NPS), (FRME), (ROW), (COL), LMNP, LMEF
Calls: DSPM

4.4.9.13 IZTQ Initialize Text Queue:

Initializes the first location of the text film copy queue, TQCT (Text Queue Count) and TAFQ (Text Access Full Queue) from A Register. Resets pointers TAQT (Text Access Queue Top) and TAQB (Text Access Queue Bottom) to equal TAQS (Text Access Queue Start).

Uses: TAQS, TQCT, TAFQ, TAQT, TAQB
Calls: None

4.4.9.14 LPR Last Page Routine:

Updates User Text Parameters CRPG from NPS, FRME, ROW and COL from LFR. Calls PDQ to format and queue the 3 word text display command sequence.

Uses: (NPS), (CRPG), (LFR), (FRME), (ROW), (COL)
Calls: PDQ

4.4.9.15 MGR Magnify Routine:

Formates and queues 3 word text display command sequence from user text parameters FICH, TFAD, FRME AND SECT. Stores in user queue at DQ and calls FTR to flag the text request.

Uses: (FICH), (TDAD), (FRME), (DQ), (SECT)
Calls: FTR

4.4.9.16 MTAP Move Text Access Parameters:

Causes packed input character string of 10 characters at PADR (User line buffer for CTSS messages) to be unpacked into a block at PKBS. Replaces last character with ETX (03) and saves pointer to last output word in user parameter KBS.

Uses: (PKBS), (PADR), (KBS), X3
Calls: UPK

4.4.9.17 NPR Next Page Routine:

NPR sets BR = 1 and calls QDR (Queue Display Request) to update user text parameters and format and queue the text request command sequence for the given user request.

Calls: QDR

4.4.9.18 PDQ Put Display Queue:

Formats and stores in user display Queue (DQ) a three word text display command from user text parameters FICH, TDAD and FRME. Calls FTR to flag the text display request. Each user can queue only one display request command sequence in DQ. If he generates a new request, it replaces any unused request.

Uses: (FICH), (DQ), (TDAD), (FRME)
Calls: FTR

4.4.9.19 PPR Previous Page Routine:

PPR sets BR = 0 and calls QDR (Queue Display Request) to update user text parameters and format and queue the text request command sequence for the given user request.

Calls: QDR

4.4.9.20 QDR Queue Display Request:

Calls IDFR (Increment-Decrement Frame) to update user text parameters according to the given text request which is indicated by the contents of BR when QDR is called. Next QDR calls PDQ (Put Drum Queue) to format and queue the three word text display command sequence for the given text request.

Calls: IDFR, PDQ

4.4.9.21 QPOP Queue Pop:

Calls CTC (Core to Core) to move text film requests up in the film request queue area so that the oldest request is at the top space. Updates the pointers TAQT (Text Access Queue Top) and TAQB (Text Access Queue Bottom). Restores the X register before returning.

Uses: ZX, TAQB, TAQS, TAQT
Calls: CTC

4.4.9.22 SPR Same Page Routine:

Calls PDQ to format and queue a three word text display command sequence from user text parameters FICH, TDAD and FRME.

Calls: PDQ

4.4.9.23 TDIG Text Digit:

Tests contents of A register to be BCD number with codes between (inclusive) 060 and 071 (octal) corresponding to decimal digits 0- through 9. Normal return with character

still in AR. Return with AR = 0 indicates AR out of specified range.

Uses: TDGT
Calls: None

4.4.9.24 TFAT Test For Automatic Text:

Tests TDSI (Text Display Indicator) to determine if a CTSS generated text request is being processed. If so, calls ETR (Enter Text Routine) to format and queue text display request.

Uses: (TDSI)
Calls: ETR

4.4.9.25 TFE Test For End:

Tests given 7 bit number for End of Text (003), Line-feed (012), carriage-return (015) or Space (040). Enter with number in AR right justified. If no end is found return is made with the given number in the AR. If an end is found return is made with AR = 0.

Uses: SVT
Calls: None

4.4.9.26 TQPU Text Queue Put:

Tests for space on the bottom of the text film request queue by comparing TAQB (Text Access Queue Bottom) to TAQE (Text Access Queue End). If there is no room at the bottom of the queue TAQT (Text Access Queue Top) is compared to TAQS (Text Access Queue Start) to determine if there is room at the top of the queue space. If there is room at the top QPOP (Queue Pop) is called to move the queue entries up to fill the empty places at the top and make room at the bottom. The film queue is a first-in first-out (FIFO) queue. Items are entered at the bottom and are processed from the top. Items are moved up (popped-up) in the queue space only if there is no room at the bottom for new entries. If there is no room at either top or bottom of the queue the flag TAFQ (Text Access Full Queue) is set, a queue full message is displayed on the user's label line and the last copy request is ignored. If there is room in the queue a three word text film request is constructed from user parameters TFAD (Text Film Address), FFRM (First Frame), FICH (Fiche), NPS (Number of Pages) and placed in the text film queue using pointer TAQB (Text Access Queue Bottom) and TQCT (Text Queue Count) is incremented.

Uses: (FICH), (TFAD), (FFRM), (NPS) TAQB, TAQE, TQCT, TAQT, TAQS, TAFQ, LQFM

Calls: QPOP, DSPM

4.4.9.27 TQTK Text Queue Take:

If the top location in text film copy queue is not zero, a three word command sequence from the copy queue is output to the Text Access Central Station interface. TAQT (Text Access Queue Top) is updated and TARP (Text Access Request Pending) is set to all ones. TQCT (Text Queue Count) is decremented and TAFQ (Text Access Full Queue) is reset to zero. If the queue is now empty IZTQ (Initialize Text Queue) is called to initialize the text queue.

Uses: TAQT, TARP, TQCT, TAFQ
Calls: IZTQ

4.4.9.28 X10 Times 10:

Multiplys the given binary number by 10 (decimal). Enter with the number in AR, return with the product in AR.

Uses: T1
Calls: None

4.4.9.29 ZT Zero Text:

Sets Text Access Request Pending (TARP) flag to zero and initializes text hardware interface. Used in case of system malfunction when text replies do not occur to clear text request that is pending according to program flags.

Uses: TARP
Calls: None

4.4.10 Drum Utility Routines

4.4.10.1 RWDL Read Write Drum Lines:

Enter with Drum Command Sector Word in AR, Track Word in BR and number of lines in XR. Waits for Drum Free condition, tests for XR = 0 then outputs Drum Commands to Drum Control Unit. Increments Sector Word for next line, decrements number of lines and loops to wait for drum.

Calls: None

4.4.10.2 RWDT Read Write Drum Track:

Enter with Drum Command Sector Word in AR, and Track Word in BR. Loads line count of 32 (decimal) into XR and calls RWDL (Read Write Drum Lines) to read or write a track of 32 lines. Increments Sector and Track Words and returns.

Calls: RWDL

4.4.10.3 RWI Read Write Initiate:

Enter with Drum Command Sector Word in AR, Track Word in BR and number of tracks in XR. Saves contents of XR in TRCT and calls RWDT (Read Write Drum Track) to read or write one track. Decrements track count and if not zero loops for another call to RWDT. Returns when track count in XR is zero.

Uses: TRCT
Calls: RWDT

4.4.11 Teletype Routines

4.4.11.1 DMPO Dump Octal:

Prints on the TTY an octal dump of given core locations. DMPO formats the output page.

Uses: TAD (Through Indirect Pointer at 0267)
Calls: OCRL, SPA5, ODMP, RPTO

4.4.11.2 FF5L Form Feed 5 Lines:

Calls RPTO to output 5 CR/LF pairs to the TTY.

Calls: RPTO

4.4.11.3 LD Load Dump:

Saves Octal Dump parameters and calls DMPO to dump a page. Increments and tests page count. If not done,spaces to new page and loops for another page.

Uses: TAD, PCT, WCT
Calls: DMPO, FF5L

4.4.11.4 NULL Null:

Turns the TTY Punch ON, outputs 64 (decimal) null characters with odd parity. Then turns TTY Punch OFF.

Uses: L222, L100, L200, L224
Calls: SWR

4.4.11.5 OATY Output A on TTY:

Waits for TTY print character buffer free and outputs contents of A Register to TTY.

Calls: None

4.4.11.6 OCRL Output Carriage Return Line Feed:

Loads A Register with CR/LF pair and calls OTCH (Output Two Characters) to output the pair to the TTY.

Uses: CRLF
Calls: OTCH

4.4.11.7 ODMP Octal Dump:

Enter with binary number in BR. Outputs a six octal character representation of the binary number to the TTY.

Calls: OATY

4.4.1.18 OTCH Output Two Characters:

Waits for TTY Print Character Buffer to become free and then outputs the two characters packed in the AR to the TTY. The left side (most significant) character is output first.

Calls: None

4.4.11.9 RPTO Repeat Output:

Enter with count n in BR and character right-justified in AR. Calls OATY to output the same character n times.

Calls: OATY

4.4.11.10 SPA5 Space 5:

Loads the number 5 into the BR and the Space Character into the AR. Calls RPTO to output the Space five times.

Calls: RPTO

4.4.11.11 SWR Send Word:

Waits for TTY Print Character Buffer to be free and then outputs contents of AR to TTY.

Calls: None

APPENDIX A

BRISC TERMINAL FEATURES AND CONTROLS

A diagram of the console is shown in Fig. A.1. The current BRISC features the following controls:

A. PUSHBUTTONS and INDICATOR LIGHTS located across the top of the keyboard (reading from left to right):

1. The red POWER ON light/button which should be lit when the console is to be used.

2. The red SYSTEM DOWN light which indicates that the system is not in operation. (The system may not be in operation even though the light is not lit.)

3. Two red indicator lights which are not currently used.

4. A red NEW PAGE button that is used to obtain a new display page. It should be used during a catalog search when the current page becomes full (see END OF PAGE light described below) and the user wishes to continue the search dialog on a new page. This button selects the next display page, erases the contents of that page and signals the main computer that it may continue sending information. If the user is on the last display page (page 14) when he presses NEW PAGE, the system displays his first page (page 1) with the message *PRESS ERASE TO CONTINUE* THIS IS DIALOG PAGE 1 appearing on the last line of the catalog display. If he then presses the ERASE button (described below), page 1 will be erased and the main computer signaled to continue, starting at the top of the page. Under these conditions if the user presses NEW PAGE instead of ERASE the dialog would continue on page 2 instead of page 1. The above message only appears when the user requests NEW PAGE when he is currently located on page 14. The NEW PAGE BUTTON is only active when the console is in Catalog Mode. (See CAT MODE below.)

5. A green END OF PAGE light indicates that the bottom of the screen has been reached. When this happens the user will normally press the ERASE button and continue using the same display page. When the

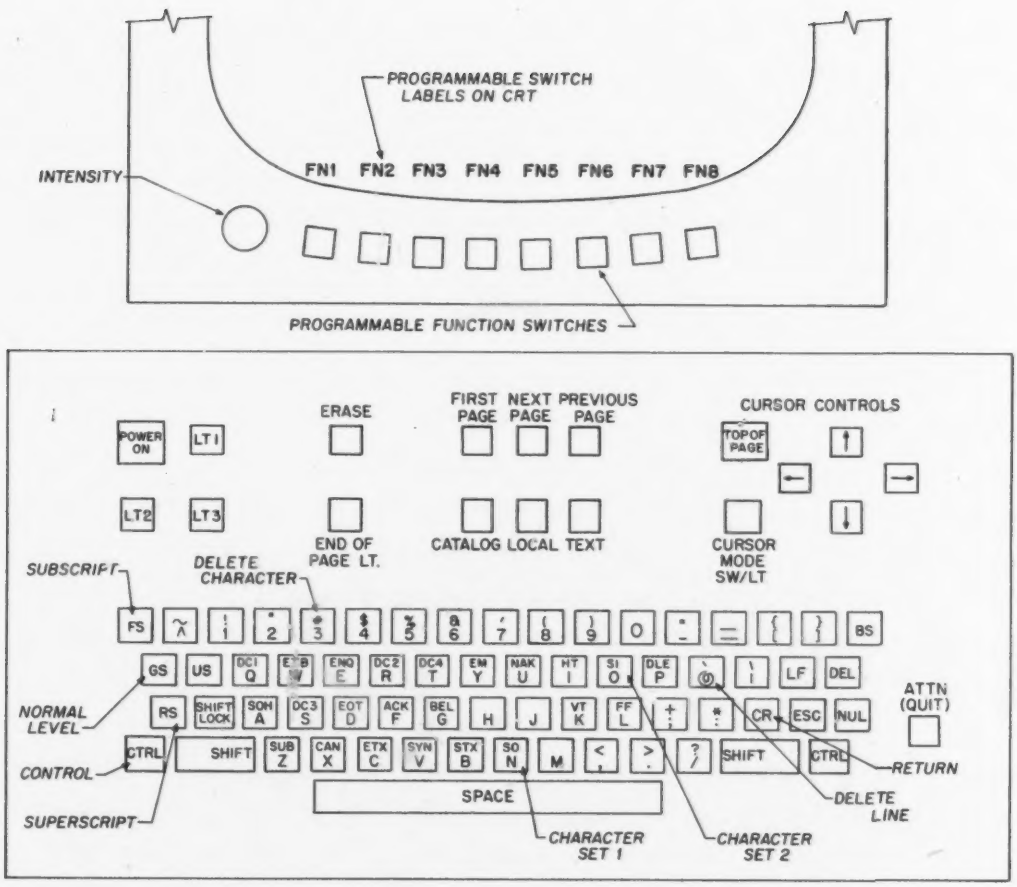


Fig. A.1 BRISC-Catalog Station Keyboard and Function Switches

END OF PAGE condition occurs the user must press either the NEW PAGE button or the ERASE button before the dialog can continue.

6. Three green paging buttons marked lst (First) PAGE, NEXT PAGE, PREV (Previous) PAGE turn "pages" on the catalog screen. Use of these buttons allows the user to page back over information received earlier in the search dialog. The user has 14 display pages at his disposal and if he presses NEXT PAGE when on page 14 he is returned to page 1. Pressing PREV PAGE when on page 1 places him on page 14. Pressing lst PAGE places the user on page 1.

7. Three black mode buttons are located beneath the paging buttons. They are:

a. CAT (Catalog) MODE: Pressing this button puts the console in Catalog Mode which allows the user to conduct catalog searches, obtain full text display and employ the Note feature of the console. Entering Catalog Mode places the console in a non-cursor mode in which the Cursor Symbol does not indicate where characters typed on the keyboard will appear on the screen. (See Edit Mode) In Catalog Mode the Cursor Symbol is normally used when employing the Note feature of the console. Pressing the CAT MODE button causes the label line (last line) of the display to be rewritten. The label line in Catalog Mode indicates which of the 14 display pages available to the user is in current use. The label line also provides the following labels for the buttons located just below the screen:

(i) ERASE: Pressing this button causes the contents of the display page being viewed to be erased. That erased information is no longer available in local storage and if needed again must be obtained from the central computer unless it has been stored by the user on the Note Page (or Save Page). In Catalog Mode the use of the ERASE button signals the main computer that the console is ready to receive information. (See END of PAGE) If the user begins typing at the keyboard immediately after an erase, the characters typed will appear at the top left of the screen. If for some reason the search information supplied by the system appears to be writing off the bottom of the screen, use of either the ERASE or the NEW PAGE (see below) button after the writing has ceased, will usually cause the next batch of information to fit properly on the page.

(ii) NOTE: Pressing this button 'notes' (stores) the line of display information on which the blinking cursor symbol appears, on a special working page called the Note Page. The Note Page is distinct from the 14 display pages available to the user. The purpose of the Note feature is to allow the user to take notes of information he may need later in the session. He may store up to 30 lines of information on the Note Page. By selecting information and storing it on the Note Page he may then erase and re-use his display pages for further dialog.

To store a line on the Note Page the user positions the blinking Cursor Symbol, by use of the Cursor Controls, on the line he wishes to note and then presses the NOTE button.

The NVIEW and NCLEAR buttons described below are used in conjunction with the NOTE button.

(iii) NCLEAR (Note Clear): Pressing this button initializes the note-taking mechanism so that when the user presses the NOTE button described above, the noted line of information will be placed on the top line of the Note Page. The user presses NCLEAR at the beginning of a session and may then note up to 30 lines by pressing the NOTE button once for each line. NCLEAR is employed only when the user wishes to restart note storage at the top of the Note Page.

(iv) NVIEW (Note View): Pressing this button causes the Note Page to be displayed so that the user may inspect the contents. The following labels appear on the last line of the display screen: ERASE, SVLNE, SCLEAR, RETURN, VDEMO, ERLNE, EROP, and RTLNE. To return to the display page on which the user pressed NVIEW, he presses the same button which now has the label RETURN. The other buttons are described under EDIT below. The user may employ these buttons to edit the Note Page. Especially he may wish to ERASE the Note Page as well as use the NCLEAR when starting a session. NCLEAR merely establishes that the next line 'noted' will be stored on the top line of the Note Page. NCLEAR does not remove previously 'noted' information from the Note Page. Pressing NOTE replaces previous information a line at a time.

b. EDIT: Pressing this button puts the user in the Edit Mode of operation which allows modification of the information appearing on the display pages. The user may make annotations or comments on the

pages, or may rearrange and store away information for later usage. Entering Edit Mode puts the console in Cursor Mode and causes the following labels to appear over the buttons located at the bottom of the screen:

(i) ERASE: Pressing this button causes the contents of the display page being viewed to be erased. The erased information is no longer available in local storage and if needed again must be obtained from the central computer. If the console has been placed in non-cursor mode characters typed immediately after, an ERASE will appear at the top left of the screen. In Cursor Mode characters typed appear where the blinking cursor symbol is displayed.

(ii) SVLNE (Save Line): Pressing this button causes the line of information on which the blinking cursor symbol appears to be 'saved' on a special Save Page (the same physical page as the Note Page of the Catalog Mode). The Save Page is a working space available to the user in addition to the 14 display pages. Consecutive pressing of the SVLNE button causes the lines indicated by the cursor to be saved on consecutive lines of the Save Page. Up to 30 lines of information can be saved in this manner. The Cursor Control buttons are used to move the blinking Cursor Symbol and so indicate which line is to be saved when the SVLNE button is pressed. The console must be in Cursor Mode for this operation. Entering the Edit Mode places the console in Cursor Mode and this is indicated by the CURSOR MODE light being lit. If the CURSOR MODE Light is not lit, pressing the CURSOR MODE Light (which is also a button) will light it and put the console in Cursor Mode.

(iii) SCLEAR (Save Clear): Normally when the user wishes to begin using the Save Line feature for the first time during his search session he initializes the saving mechanism by pressing the SCLEAR button. This makes the SVLNE, RTLNE and NOTE functions operate from the top line of the Save (or Note) Page. Normally the user presses the SCLEAR at the beginning of his session and does not use it again until he has saved 30 lines of information. If he wishes to save more than 30 lines he must move the information from the Save Page. See RTLNE below.

(iv) SVIEW (Save View): Pressing this button displays the Save Page to the user so that he can see what is stored on that page. At the beginning of the session he may want to press SVIEW to obtain the

Save Page display and then press ERASE to remove anything that a previous user may have left on the Save Page. It is not necessary for the user to erase the Save Page since the Save Clear function starts operations from the top of the Save Page and as lines are saved they replace line by line the information that was previously stored on the Save Page. Only the unused portion of the Save Page may contain information stored there at an earlier session. Thus for neat housekeeping the user may wish to erase the Save Page when starting.

Pressing SVIEW causes the following labels to appear on the last line of the screen: ERASE, SVLNE, SCLEAR, RETURN, VDEMO, ERLNE, EROP, and RTLNE. The user presses RETURN to get back the Display Page on which he pressed SVIEW. Pressing VDEMO (View Demonstration) displays a demonstration page whose contents illustrate special features of the console and system. The user returns to his display page by pressing RETURN which is the only labeled button on the Demo Page. In the Viewing Mode most of the mode and paging buttons are not active. Normally the user will employ the buttons whose labels are on the last line of the screen when he is in View Mode. Especially useful are RETURN and ERASE.

ERASE, SVLNE, SCLEAR, ERLINE, EROP, and RTLNE are described below:

(v) *edit*: Pressing this button causes the console to reenter the Edit Mode (even though it is already in Edit Mode). This causes the label line to be rewritten and Cursor Mode to be entered. Under certain circumstances this action is useful. However, it is not a normal usage. The * and the lower case letters 'edit' indicate that the console is in Edit Mode.

(vi) ERLNE (Erase Line): Pressing this button erases the line upon which the Cursor Symbol is displayed. Normally the user positions the blinking Cursor on the line to be erased by using the Cursor Controls and then presses the ERLNE button.

(vii) EROP (Erase Rest of Page): Pressing this button causes the portion of the Display Page starting, with the line on which the Cursor is positioned, to the bottom of the page (excluding the label line) to be erased.

(viii) RTLNE (Return Line): Pressing this button causes a line of information to be returned from the Save Page and written on the Display Page on the line indicated by the Cursor. The SCLEAR (Save Clear) button initializes the Return Line mechanism so that after pressing SCLEAR, when the user first presses RTLNE, the first line of information stored on the Save Page is returned to the Display Page and written on the line where the Cursor is positioned. The second time RTNE is pressed, the second line of the Save Page is returned. The normal mode of use is to press SCLEAR at the beginning of a session. Next by consecutive use of the SVLNE button, information is saved on the Save Page. At any time that information may be viewed by pressing the SVIEW button followed by the RETURN (different from the RTLNE) button to get back to the Display Page. There is no real need to use the RTLNE button until the Save Page is full. At that time the user may page to some conveniently located page such as Page 1 or Page 14, ERASE that page, position the Cursor at the TOP of the page, press RTLNE, move the Cursor down a line, press RTLNE, etc. until the desired information has been returned from the Save Page. The lines are returned from the Save Page in the same order in which they were saved. It should be noted that the SCLEAR button initializes both the SVLNE and the RTLNE mechanisms. Information is not removed from the Save Page by pressing SCLEAR, however, pressing SCLEAR and then pressing SVLNE replaces the first line of the Save Page with new saved information.

In Edit Mode the NEW PAGE button, and the Text Access buttons on the Text Access Panel are not active.

c. TEXT MODE: Pressing this button allows the user to obtain full text display without reference to the main computer. This is not the normal way to obtain text display and requires that the user know where the desired text is located in the microfiche storage. The label line informs the user that he has entered Text Mode with *text access* written on the last line of the screen. The following labels also appear on the last line: TCOPY, TFIRST, TLAST, TCLEAR.

To operate in Text Mode the user types the microfiche access number on the keyboard in the format: fiche number, space, starting row and column, space, number of pages and carriage return. The message *TEXT SYSTEM HANDSHAKE* appears on the last line of the display to

indicate that communication has been established with the text system. This should be followed by the appearance of the microfiche image on the associated text storage-tube display. This is a separate screen from that on which the catalog information is presented. Next the message *TEXT SYSTEM READY* appears on the last line of the catalog display to indicate that the user may make another text request either by typing another access number of by using any of the text panel buttons described below.

Pressing TCOPY (Text Copy) requests a film copy of the last microfiche document which was displayed. First the user obtains the microfiche display on the storage tube and then presses the TCOPY button. The Adviser should be consulted before making use of the Text Copy feature.

The TCLEAR (Text Clear) button is a program Reset button which is meant to be used by the adviser when the Text Handshake message does not appear after making a text display request and waiting for two minutes (someone else may be employing the Text Copy feature and so have the text system busy for that time). It causes the system to "forget" about any unanswered text requests and accept new requests.

The TFIRST (Text First Page) and TLAST (Text Last Page) buttons below the last line of the Catalog display screen when in Text Mode have the same function as the buttons labeled in the same manner on the Text Access Panel, (see description below).

The buttons located on the Text-Access Control Panel may be used in either Text Mode or Catalog Mode.

The Cursor Controls, a set of white buttons, are used to control the location, visibility and function of a position marker or cursor displayed on the catalog screen. Visibility of a blinking cursor symbol (the ampersand) is controlled by the CURSOR MODE button/light. If the light is lit, the console is in the Cursor Mode of operation and the Cursor symbol should be visible. If the CURSOR MODE light is lit and the Cursor symbol is not visible, press the Cursor control button marked TOP and the symbol should appear at the top left corner of the catalog display screen. If the symbol does not appear, consult an Intrex adviser.

Pressing the CURSOR MODE button when the CURSOR MODE light is lit takes the console out of Cursor Mode and turns the CURSOR MODE light off.

(Depending upon how the Intrex Adviser has set the console, this may or may not make the Cursor Symbol invisible.)

Pressing the CURSOR MODE button when the CURSOR MODE light is not lit puts the console into Cursor Mode, turns the CURSOR MODE light on and displays the Cursor Symbol.

When the console is in Cursor Mode, the Cursor Symbol indicates the position on the screen where the next typed character will appear. The Cursor may be moved in the indicated direction by pressing the white buttons labeled →, ←, ↓, ↑. Pressing one of these buttons and releasing it promptly moves the cursor symbol one position while holding it down causes the symbol to 'run' across the screen. Pressing the white button labeled TOP puts the Cursor Symbol at the top left position of the screen.

Entering the EDIT Mode by pressing the EDIT button places the console in Cursor Mode, lights the CURSOR MODE light and displays the Cursor Symbol. (See Edit Mode.)

Entering Catalog Mode by pressing the CATALOG MODE button places the console in non-cursor mode and turns the CURSOR MODE light off. The Cursor Symbol may or may not be displayed. Normally Catalog Searchers are conducted in a non-cursor mode and the Cursor Symbol may be used with the NOTE feature described above.

B. ADDITIONAL KEYBOARD AND DISPLAY FEATURES

1. The ATT/QUIT button (black) is located to the right of the keyboard and is used for interrupting system output.

2. The carriage return button is marked CR. It is located toward the right side of the third row of keys from the top of the keyboard.

3. The Intensity Control knob on the lower right beneath the screen can be used to control the intensity of the display.

4. The programmable buttons are located beneath the tube face and the labels for them appear on the tube face immediately above the relevant button. Their functions are described in Part A above.

5. The Text Access display screen and control panel shown in Fig. A.2 include the following buttons and lights:

a. Five light/buttons for paging through the full text contained in microfiche storage. These are FIRST PAGE, NEXT PAGE, PREV. PAGE, SAME PAGE and LAST PAGE. These buttons may be pressed in Catalog and Text Modes but are inactive in Edit Mode. On the BRISC-TEXT PANEL, the buttons may be used whether they are lit or not. Centering of the page image on the screen cannot be adjusted by the user. However, use of the SAME PAGE button may improve the positioning when the image is rewritten. If that does not correct the problem, the Intrex Adviser should be consulted.

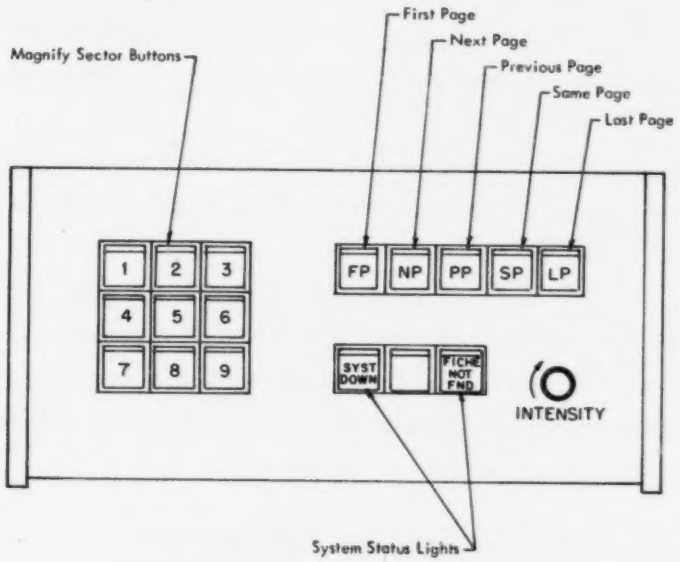


Fig. A.2 Text Access Control Panel

b. Nine Sector Buttons used to Magnify the Text image displayed on the Text screen.

The SECTOR MAGNIFICATION buttons (marked 1 through 9) select a sector of the page image for magnification. Roughly one quarter of the page is magnified and the position of the SECTOR button indicates the location of the sector to be magnified. Only one level of magnification is possible (double the size in each dimension). Repeated use of the magnification features yields the same degree of magnification. Usually the total page may be magnified by use of the sector buttons marked 1, 7, 3 and 9.

c. Yellow Status lights indicating the status of the text subsystem and displaying messages associated with special conditions encountered as a result of the last text display request. These messages also appear on the last line of the BRISC Catalog display.

SYSTEM DOWN: indicates that the Text subsystem, but not necessarily the Catalog system, is not in operation.

ERROR: indicates when a system error has occurred (e.g., an improper fiche number has been received). The user should repeat his request when this light lights.

FICHE NOT FOUND: indicates that the requested fiche was not found. This may be either because it is not in the fiche storage or a malfunction has occurred. The request should be repeated.

END OF DOCUMENT: indicates that the user has pressed NEXT PAGE while his current page is the last frame (E12) on the selected fiche, or has pressed PREV. PAGE while viewing the first frame (A1) of the fiche. (The light actually indicates the ends of the fiche rather than the ends of the document.) On the BRISC an attempt to page past the actual DOCUMENT limits produces a corresponding message on the label line of the Catalog display. The Text Panel lights will go out, however, they are still active and may be pressed even though they are not lit.

On all Text Consoles the request buttons light up and when pressed the lights go out until the text display sweep is complete at which time they re-light. In general, a button may be pressed when it is lit, however, on the BRISC, the buttons are active even when they are not lit if the BRISC console is in either the Catalog Mode or the Text Mode of operation. On the other hand, when the BRISC console is in any mode other than Catalog or Text, the Text Panel buttons are inactive whether lit or not.

d. An image intensity control. If the image on the screen does not have the proper degree of brightness for reading, the user can adjust it by using the brightness control. The image is stored on the screen so that adjustment of brightness will not change the brightness of the displayed page until the page is rewritten or another page is displayed.

APPENDIX B

MAGNETIC DRUM STORAGE — Principles of Operation

Contents of Appendix B

Magnetic Drum Storage-General Description

Drum Sector Addresses

Drum Storage for the Intrex Display System

Relation of Sector Addresses to CRT Display Characters

Drum Track Initialization Prior to CRT Viewing

Operations of the Drum Control Unit

Drum Commands

Drum Command Codes (Used in the Sector Word)

Write Single Character Command Word Format

Read/Write Drum Line Command Word Format

Select Drum Track for Viewing Command Word Format

Read Drum Sector Counter

Read Drum Revolution Counter

620i Instructions

Drum Unit Device Addresses and Function Codes

Write Single Drum Character Example — Command Format and 620i Program Segment

Read/Write Drum Line Example — Command Format and 620i Program Segment

Timing Considerations for Read/Write Drum Line Operations

Select Drum Track for Viewing Example — Command Format and 620i Program Segment

Read Sector Counter — 620i Program Sequence

Read Revolution Counter — 620i Program Sequence

620i I/O Device Interrupt Capability

I/O Device Interrupt and Trap Priorities

Drum Unit Priority Interrupt

Enable/Disable Drum Interrupts

Master Interrupt Enable/Disable Switch

Initialize Drum Unit and 620i Manual System Reset

Note on Drum Interrupt Terminology Related to Interrupt Switch and Interrupt Mask

Note on Character Codes within the Intrex Display System

Summary of Drum Controller/620i Addresses and Codes

Collection of Drum/CRT Terms, Parameters and Specifications

This Appendix presents general information about the Magnetic Drum Storage Unit incorporated into the Buffer-Controller of the Intrex Display System. The report also discusses specific features of the drum control unit that are of importance to system programmers of the Buffer-Controller 620i processor when called upon to program drum operations.

Magnetic Drum Storage — General Description

A Vermont Research C747 magnetic drum unit and an associated Drum Controller are incorporated into the Buffer-Controller of the Intrex Display System. The Drum Controller was designed by R. Goldschmidt and fabricated at the Electronic Systems Laboratory of MIT.

The current drum storage capacity is 655,360 bits. The storage is organized into 32 tracks with each track containing 2048 ten bit sectors. The first and last bits of each sector are employed as framing bits by the hardware timing mechanism while the remaining eight bits are available for data storage. Each sector may provide storage for an eight bit character (or for an eight bit data byte) and each drum track has a 2048 character capacity. The current system of 32 tracks then has a data capacity of 65,536 data bytes (or 8 bit characters). Every sector is addressable through the Drum Controller with its associated sector counter.

The drum unit provides the Drum Controller with three primary timing tracks and three backup timing tracks. The timing tracks supply bit, sector and revolution time signals.

The drum revolves at 3528 rpm. This corresponds to 17 ms (milli-seconds) per revolution. The maximum latency time in reading or writing is 17 ms and the average is 8.5 ms. Each of the 2048 characters pass beneath the read/write head for a given track during one drum revolution. The drum control unit can read or write on only one

drum track at a time. The data transfer is serial bit by bit between the drum and the control unit and the transfer time for each sector is approximately 8.3μs (micro-seconds). This corresponds to a sector frequency of 120,480 sectors (or characters) per second.

The drum unit is interfaced with the I/O bus of a general purpose digital computer,the Varian Data Machines 620i. This small processor is incorporated into the Buffer-Controller of the Intrex Display System. The drum storage may be used as general purpose storage for the 620i processor or as refresh storage for the CRT displays of the Intrex Display System. The displays are character oriented and drum sectors are used as storage for eight bit character codes when supplying CRT refresh data. The 620i core storage is addressable in 16 bit computer words. When drum storage is used for 620i program or data secondary storage,two drum sectors are taken to store each 620i sixteen bit word.

Drum Sector Addresses

Each of the 2048 drum sectors (or character positions) of a track are addressable by the drum control unit through the mechanism of a Sector Counter. The sectors are addressed 0000 to 2047 (00000 to 03777 in octal notation). The sectors are addressed in binary and it is convient to consider the 11 bit sector address as a 5 bit line number concatenated with a 6 bit character position. The low order 6 bits are taken as the character position and the remaining 5 bits as the line number. Character positions are numbered from 0 to 63 (decimal) and lines 0 to 30 (decimal). A sector address of 65 (00001000001 in binary) corresponds to character position 1 of line number 1 which refers to the 2nd character on the 2nd line of a given track.

The drum tracks are also addressed through the drum control unit. In the current system there are 32 tracks addressed 0 to 31 (decimal).

Drum Storage for the Intrex Display System

The primary function of the magnetic drum incorporated into the Buffer-Controller of the Intrex Display System is to supply refresh information to the CRT (cathode ray tube) display of the BRISC (Buffered

Remote Interactive Search Console) terminals. Characters typed on the keyboard of a console become visible on the user's CRT when the corresponding character code is written on the drum track which refreshes the page display viewed by the user. A general purpose digital computer, the Varian Data Machines 620i, also incorporated into the Buffer-Controller, supplies the drum control unit with commands for reading and writing on the drum surface. The 620i can select one of a set of drum tracks and assign that track to a console for use as refresh storage for the CRT of that console. Once selected the data from that drum track automatically refreshes continuously that CRT. Changing the CRT display page contents corresponds to changing the data content of the associated drum track. The 620i program controls the modification of drum track data content.

A single drum track corresponds to a single display page on the CRT. The 620i stored program can write 2048 eight bit characters on each drum track. Only 1736 of these character positions are visible to the viewer of the CRT screen. Details of the relationship between the character code position on drum storage and the character display position on the CRT screeen are presented in the next section of this report.

The CRT display is an alphanumeric display in which the patterns made visible on the CRT screen are generated by a dot matrix device with a fixed set of 192 characters. The display presentation has a fixed format of 31 lines with 56 alphanumeric character positions per line. Thus the CRT page has a capacity of 1736 characters. The CRT is refreshed from drum storage and for each character position on the page there corresponds an addressable sector or character position on the selected drum track.

Relation of Sector Addresses to CRT Display Characters

Each sector address represents a fixed physical position on the drum track. That physical space is called a sector and has a ten bit storage capacity. Two of the ten bits are used as framing bits by the drum control circuits leaving eight bits of each sector for data storage. When the drum track is being used to refresh the CRT display the eight data bits contain character codes from the ASCII character

set. The Buffer-Controller supplies the string of eight bits representing an ASCII character to the character generator of the display console which programs the CRT circuitry to display the correct dot pattern to present the character image on the screen.

By 620i Buffer-Controller convention a synchronization character code is stored on the drum sector with the highest sector address (03777 octal). The sector counter is a modulo 2048 (decimal) counter and the sector location which follows that having the highest address is the one which has the lowest address (00000). As the drum revolves character codes are read from the drum sectors in the order of increasing sector addresses. Detection of the synchronization (SYN) character controls the display page sweep retrace circuitry causing the contents of sector address zero to become the first ASCII character to be displayed at the top-left position of the CRT screen. Successive drum sectors supply codes for successive characters on the first line of the CRT display until 56 characters have been presented on the first display line. The contents of the next eight sectors are not displayed on the CRT and the time required for them to pass beneath the drum read head is used to reposition the CRT beam to the beginning of the second display line. Then the contents of the 65th (decimal) sector (sector address 64) is used to generate the first character of the second display line. This format is repeated for each display line.

The 2048 sectors per drum track may be thought of as organized into 32 lines with each line composed of 64 eight bit characters. Only 31 lines can present visible characters on the CRT display and each line can display only 56 visible characters. The position of the synch (SYN)character on the drum track establishes the correspondence between drum storage locations and CRT display positions. The invisible eight character positions at the end of each line correspond to valid drum storage sector addresses and data may be stored there. The same comment holds for the sectors corresponding to the 32nd line which is also not visible on the CRT. The time required for this last drum line to pass the read station is used by the CRT hardware for the vertical retrace. Details of the software conventions related to the last drum line are discussed in the section on drum track initialization which follows.

Drum Track Initialization Prior to CRT Viewing

The Intrex CRT display hardware obtains a synchronization signal from the refresh data furnished by the drum unit. This signal causes the CRT sweep circuitry to reset and release a new display sweep from the top left of the CRT screen. The synch signal is coded as a special character (the SYN character) which has been stored on the drum track furnishing the CRT refresh data. For correct operation the console hardware requires a two character sequence, a delete character followed by the synch character. In the seven bit ASCII code the delete (DEL) character is coded 0177 (octal) and the synch (SYN) character 026 (octal).

The two character sequence may be placed in any consecutive pair of sectors on the refresh track. The CRT display characters are referenced to that sector pair as immediately preceeding the first character of the first CRT display line. By 620i software convention the DEL character is placed in character position number 62 (decimal) and the SYN character in position number 63 of the last drum line, line number 31 (decimal). Thus the SYN character occupies the last character position on the last line of the refresh track.

The console decodes certain ASCII control characters in addition to the DEL and SYN characters as they are received from the refresh track. All 8 bit codes correspond to some ASCII character; however, some represent control characters having special effects on the console hardware. Attempting to view arbitrary binary data on the CRT results in the console interpreting certain bit combinations as control characters and usually producing erratic displays on the CRT. To prevent this the refresh track should be initialized by the 620i program writing known alphanumeric characters in all 2048 sectors except for those holding the DEL and SYN characters. Both the Space (040 octal) and the Null (000) characters are valid ASCII alphanumeric characters and may be used for drum display page initialization. These two characters are 'invisible' characters, that is they present no dot pattern display on the CRT screen, but they do have unique codes and occupy space on the drum track. By convention the 620i Intrex programs initialize drum tracks with the Null character and certain alphanumeric system messages depending upon the console user's program status. Details of the Intrex

software initialization procedure for CRT refresh pages will be given in the memo on Intrex display console principals of operation.

Operations of the Drum Control Unit

The drum control unit interfaces with the 620i processor of the Intrex Buffer-Controller as an I/O device attached to the processor I/O-bus (also called the E-bus). The drum unit is assigned two I/O device addresses and 620i I/O instructions may be executed using those addresses. The drum controller responds to the following 620i instructions: External Control (EXC), Program Sense (SEN), Input/Output to/from A or B Registers or core memory. The drum control unit also generates priority interrupt requests and interrupt addresses and employs the 620i direct memory access (DMA) and control trapping mechanism for block transfer of data with cycle stealing to allow overlapped operations.

The drum control unit executes a number of operations in response to command sequences received from the 620i processor. For clarity operations executed by the 620i CPU are said to be the result of 620i instructions while those operations executed by the drum control unit are the result of drum commands.

The execution of 620i I/O instructions with drum control unit addresses causes the 620i to output data to the drum control unit. That data, received in the proper sequence and under correct conditions, then become commands for the drum. Thus programming the drum control unit consists of writing 620i instructions to output the appropriate drum command sequences under proper conditions.

A list of the major operations performed by the drum unit follows:

Write Single Drum Character
Read Drum Line (64 consecutive characters)
Write Drum Line (64 consecutive characters)
Select Drum Track for CRT Viewing
Read Drum Sector Counter
Read Drum Revolution Counter

In addition, the following control operations are performed by the drum controller.

Initialize Drum Revolution Counter (reset to zero count)
Initialize Drum Unit (put drum controller in a Ready condition)
Turn On Drum Interrupt Switch (enable interrupt)
Turn Off Drum Interrupt Switch (disable interrupt)

Two additional operations have been included in the drum controller logic although the BIC (buffer interlace controller) option for the 620i needed for the execution of these operations has not been obtained for the Intrex Display System.

Connect/Disconnect BIC Read
Connect/Disconnect BIC Write

Drum Commands

The drum control unit receives read/write commands from the 620i processor. There are currently two types of read/write commands instrumented in the Intrex Display System. One is used to write single 8 bit characters on drum storage and the other is used to transfer blocks of data between drum storage and 620i core memory. The block transfer moves 32 sixteen bit words between core and drum storage. The 32 word block corresponds to one drum line of 64 character positions. The block data transfer operation makes use of the direct memory access (DMA) feature of the 620i which allows the transfer of data between the E-bus and core memory to proceed overlapped with the 620i program sequences.

The drum unit has two I/O device addresses associated with it. One is called the Sector Device Address and the other the Track Device Address. Basically the Sector Device selects the drum sector and the Track Device the drum track to be used in the operation. In some of the operations this straight-forward naming of the device addresses is not totally appropriate; however, they are convenient names. The data (commands) output to these two drum device addresses also contains other information as described below.

The Write Single Character and the Read/Write Drum Line operations executed by the drum result from a two word command sequence issued by the 620i to the drum device addresses. The first word of the sequence is the sector word which is output to the Sector Device Address. The sector word indicates the sector number of the drum

storage to be accessed by the command. It also contains the command code which informs the drum control unit which command is being issued. Additional information contained in the sector word and the exact format of that information depend upon the particular operation. These details are discussed in a later section of this report dealing with the various command formats.

The second word of the command sequence is the track word which is output to the Track Device Address. The track word contains the drum track number indicating the track to be selected for the operation. In the single character write operation the data (the 8 bit character code) to be written into drum storage is contained in the track word also. Details of the command sequence format are given below.

Drum Command Codes (Used in the Sector Word)

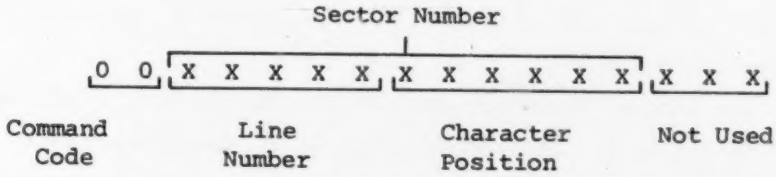
The two most significant bits of the sixteen bit sector word serve as a command code indicating how the command sequence is to be used by the drum control unit.

Code	Operation
00	Write Single Character on the Drum
01	Write a Line (64 characters) on the Drum
10	Read a Line (64 characters) from the Drum
11	BIC Control (Buffer-Interlace-Controller) this command is not operational in the current Intrex Display System.

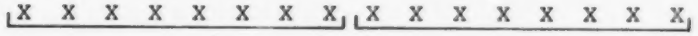
Write Single Character Command Word Format

The format of the 16 bit drum sector and track words for the Write Single Drum Character command sequence is given below.

Sector Word for Write Single Drum Character Command



Track Word for Write Single Drum Character Command



For the write single drum character command sequence the 16 bit sector word contains the following information:

Command Code — a two bit code indicating which command is being issued to the drum controller. The code 00 indicates the command write single drum character.

Sector Number — an 11 bit sector address which indicates the drum storage location where the single eight bit character code is to be written. The sector number is a concatenation of the 5 bit line number and the 6 bit character position. The sectors of a drum track may be considered organized into 32 lines each containing 64 characters. This is particularly helpful in visualizing the requirements of the character placement on the CRT display page.

The low order three bits of the sector word are not used in the write single drum character command sequence.

For the write single drum character command sequence the 16 bit track word contains the following information:

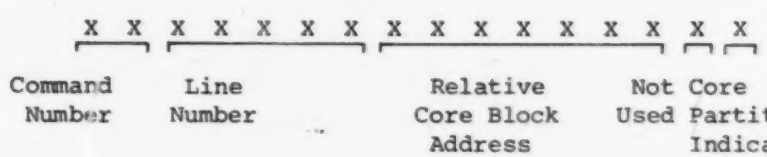
Track Number — an eight bit number indicating the drum track on which the single character is to be written. Track numbers begin with 00000000.

Character Code — an eight bit character code which is the data to be written on the drum by the operation.

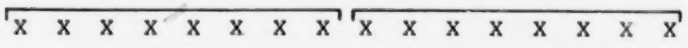
Read/Write Drum Line Command Word Format

The format of the 16 bit drum sector and track words for the Read/Write Drum Line command sequence is given below.

Sector Word for Read/Write Drum Line Command



Track Word for Read/Write Drum Line Commands



For the Read/Write Drum Line command sequence the 16 bit sector word contains the following information:

Command Code — a two bit code indicating which command is being given to the drum control unit. The code 10 indicates the Read Drum Line command and 01 indicates Write Drum Line.

Line Number — a five bit line number indicating the drum line to be accessed for the data transfer. The Read/Write Drum Line operations transfer 32 sixteen bit words (or 64 eight bit characters) between 620i core memory and drum storage. The drum line is a block of 64 consecutive drum sectors with a fixed beginning sector. Drum line 0 begins at sector number 0, line 1 at sector number 64 (decimal), line 2 at sector 128 (decimal), etc. Since each line transfer begins with the sector corresponding with a character position of 0 for the given line, the low order six bits of the sector number are not contained in the sector word for the read/write line command sequences. Only the line number (the 5 high order bits of the sector number) is shown in the sector word format for line operations. Thus for line operations the sector word contains a line number rather than a sector number.

Relative Core Block Address — a seven bit relative address which is used to determine an absolute 620i core storage address to be used as a starting address for the 32 word block transfer of data between 620i core and drum storage. The 620i has 8192 sixteen bit core memory locations and 13 bits are required to address all 8K locations. Since there are not 13 bits available for this purpose in the sector word, the core addresses used for block transfer with the drum have limitations placed upon them. For drum read/write line operations the 620i core is conceptually divided by 32 word boundries, and block transfers are aligned with these 32 word boundries. The core blocks used in drum line operations can begin with core addresses 0, 32, 64, 96, etc. This means that the low order 5 bits of core block starting addresses are always

zero and so may beleft out of the relative core address contained in the drum line sector word.

The seven bit Relative Core Address is not only given in terms of 32 word core blocks but is also presented relative to a 4K partition of core. A unique 7 bit relative core address may refer to a particular core location in either the upper or the lower 4K partition of core. To address each of the 256 blocks of 32 words requires 8 bits. The 7 bit relative core address represents the 7 low order bits of the full block address. The remaining high order bit is presented in the Core Partition Indicator described below. The relative core address taken with the core partition indicator determine the absolute core address used in the block transfer of data. An example following the description of the core partition indicator will help those interested in the details of block addresses.

Core Partition Indicator — a single bit indicator which determines the core partition to be used in conjunction with the core relative address to determine the absolute core address used as the starting location in core for the block transfer of the 32 word drum line operation. Core is partitioned into two parts, upper and lower. For an absolute core address the high order bit of the 13 bit absolute address determines if a location falls in upper or lower core. If the high order bit is 1,the location is in upper core. If it is 0,the location is in lower core. In an analogous manner if the core partition indicator is 1,then the addressed location lies in upper core and the relative core address is taken relative to the upper partition. If the core partition indicator is 0,the relative core address is taken relative to the lower core partition.

The relative core block address which is 7 bits gives a core address relative to a core partition. That is,the address is relative to absolute core location 0000 for the lower partition and relative to absolute core location 010000 (octal) for the upper partition. A relative core block address of 0000001 (binary) with a core partition indicator of 0 indicates that the address is relative to the lower core boundary at 0000. Thus the absolute core location to be used is 0000000100000 (binary) or 00040 (octal). If the relative

core block address is 0000001 (binary) as before, but the core partition indicator is 1, then the 7 bit address is relative to the upper core partition at location 010000 (octal) and the absolute location of the block starting address is 010040 (octal).

One bit of the sector word is not used in the Read/Write Drum Line command sequence.

For the Read/Write Drum Line command sequence the 16 bit track word contains the following information:

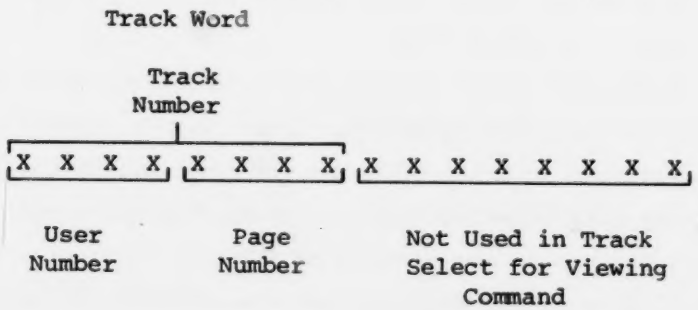
Track Number — an 8 bit track number indicating the drum track to be used in the read/write operation.

The low order 8 bits of the track word are not used in the Read/Write Drum Line command sequence.

Select Drum Track for Viewing Command Word Format

The 620i issues the drum command sequence that selects a drum track and assigns it as the refresh track for the CRT display of the Intrex display console. Once selected the track continuously refreshes the CRT until a new track is selected for that purpose. Since none of the information normally contained in the sector word is required for the select track for viewing operation, no sector word is needed in the command sequence, only a track word. The drum controller accepts a track word not immediately preceeded by a sector word as a special one word command sequence not requiring a command code nor a sector word.

The format of the 16 bit track word for the Select Drum Track for viewing operation follows:



The high order eight bits of the track word indicate the track number of the track to be selected. Not all tracks can be viewed on a particular Intrex CRT. Only 16 tracks can be viewed at a particular

console so the low order 4 bits of the track number are called the page number for a particular console. The 4 high order bits of the track number are considered a user number (or console number). Thus tracks 0 to 15 (decimal) could be viewed by user number zero while tracks 16 to 31 (decimal) could be viewed by user number one. Track 16 (010000 binary) would supply user number one with refresh information for his page number zero. User number one cannot view user number zero's pages, etc.

Read Drum Sector Counter

The Drum Sector Counter is a modulo 2048 counter. Its contents indicate the character position (sector) which is beneath the read/write heads when the counter is read. The contents of the sector counter is effectively divided by 2 on input to the 620i. This makes the counter appear to be a mod 1024 (decimal) counter and the count itself a count of 16 bit drum word positions rather than 8 bit sectors.

The sector counter is read by execution of a 620i input instruction with the sector device address 017 octal. No command sequence output is required by the drum controller for this operation. The 620i instruction itself conveys all the necessary information to the drum control unit which responds by placing a ten bit count in the low order positions of the E-bus. The remaining 6 high order bits of the E-bus are set to zero by the operation. The kind of input instruction executed by the 620i determines the destination of the data placed on the E-bus by the sector counter. The data may be taken into the A or B registers of the CPU or may be stored in the core memory.

Read Drum Revolution Counter

The Drum Revolution Counter is a mod 2^{16} (65,536 decimal) counter. Each revolution of the drum increments the drum revolution counter. The contents of the drum revolution counter are read by issuing a 620i input instruction with the track device address. No command sequence output is required by the drum controller for this device. The620i input instruction itself conveys all needed information to the drum controller.

The Revolution Counter may be initialized (set to zero) by execution of a 620i External Control Instruction or by pressing the manual System Reset switch on the 620i control panel. Details of the 620i instructions used for inputting and initialization of the Revolution Counter are given below.

The contents of the Revolution Counter are incremented once each drum revolution and so it counts in 17 ms time increments. The 2^{16} mod counter thus serves as a clock with a time range from 0 to 1,114.11 seconds (or 18.56 minutes).

620i Instructions

Commands to the drum control unit are issued by the 620i processor. The following discussion presents information about 620i instructions needed to understand drum control unit operation.

The 620i may communicate with I/O (Input/Output) devices through the I/O-bus which is also called the E-bus. Various 620i instructions cause six bit I/O device addresses to be placed on the E-bus by the 620i. The I/O device so selected may then transfer additional information to or from the E-bus. Depending upon the nature of the 620i instruction the CPU (central processing unit) transfers information between the E-bus and core memory or the general purpose 620i CPU registers.

A typical input instruction is the Clear and Input to A Register instruction.

```
CIA    017
```

Upon execution this instruction causes the contents of a general purpose 16 bit register, the A register, to be cleared to zero, the I/O unit 017 (octal) selected and the data presented by the I/O unit 017 accepted onto the E-bus and then transfered into the A register. How many of the 16 bits and which ones the I/O device places on the E-bus or takes from it depends upon the design of the I/O device and its interface. The CPU transfers all 16 bits to or from the E-bus.

The CIA operation requires two 620i cycle times (1.8μs per cycle) for completion. During the first cycle the instruction is fetched from core memory into the instruction register and the I/O

device address is placed on the E-bus to select the desired I/O device. The second cycle is required to transfer the data between the device and the E-bus and then into the cleared A register. At the end of the 3.6μs the CIA instruction is complete

A list of 620i I/O instructions follows:

CIA	Clear and Input to A Register
CIB	Clear and Input to B Register
INA	Input to A Register (OR's new data into previous contents of A)
INB	Input to B Register (OR's new data into previous contents of B)
OAR	Output A Register
OBR	Output B Register

The above instructions each require 2 cycle times (3.6μs) for completion of the operation.

Another pair of I/O instructions transfers 16 bit words between 620i core locations and the E-bus. These instructions require an extra cycle time to access core memory for the data transfer. For these instructions 3 cycle times (5.4 μs) are required for completion of the operation.

IME	Input to Memory
OME	Output from Memory

In all the above instructions the low order six bits of the instruction are used as a device address to select the desired I/O device. The drum unit has two addresses assigned to it. The sector device address is 017 (octal) and the track device address is 016 (octal).

In addition to the I/O instructions listed above the 620i and I/O devices may communicate by use of the 620i External Control and Sense instructions. External Control (EXC) allows the 620i program to output a signal to an I/O device. The Sense (SEN) instruction allows the 620i to sense conditions existing in an I/O device.

Both the External Control and Sense instructions use the 6 bit I/O device address described above. In addition 3 more bits are used

to call for a particular function within the selected device. The 3 bit function codes are placed on the E-bus by the CPU just as the device address is. Therefore when executing both the EXC and SEN instructions the 620i CPU places a nine bit function/device address on the E-bus. Usually the same device address given in I/O, External Control and Sense instructions are taken by convention to refer to a particular device for all such pertinent instructions. This is not necessary in general since the operation is also indicated on the E-bus at the same time as the device address and so could and sometimes is used for different devices in different instructions.

When it is necessary for the 620i program to signal an I/O device to begin an action, the External Control (EXC) instruction is used. An example is:

```
EXC    0517
```

Execution of this instruction selects the I/O unit 017 (octal) and presents to that selected unit the function code 5. The I/O unit will decode the function code and then initiate some device action. This operation only requires one cycle time (1.8 μs) of the CPU. The action initiated in the I/O device by receipt of the function code requires an amount of time determined by the nature of the action indicated. The CPU does not wait for completion of that I/O device operation before continuing the program sequence. If a program wait is required at that time it must be programmed in by use of the Sense instruction discussed in the following paragraphs.

To allow the 620i CPU to determine that an I/O device requires attention the program may execute the Sense (SEN) instruction which senses the condition of special addressable device lines. The condition of these lines can be either High (True) or Low (False). An external I/O device raises or lowers a particular sense line to convey a condition to the CPU. For example, an I/O device with a data buffer may indicate to the CPU that the buffer is ready for data transfer by making the corresponding sense line assume its True condition.

In the 620i the Sense instruction is a two word instruction. It is a conditional jump instruction and the second word of the instruction contains the address in core to which the program control is to

jump if the true condition is sensed on the selected sense line. An example follows.

```
SEN     0316,     CONT
```

The condition of the sense line 0316 (octal) is tested and if the condition is true the CPU takes its next instruction from core location CONT and follows the program sequence from there. If the sense line condition is false the CPU takes its next instruction from the core location following the two word SEN instruction. This allows a program branch based on the condition of the sense line.

Often the sense lines are associated with I/O devices and so are assigned the same 6 bit device address as the I/O device. The 3 bit function codes allow different conditions to be sensed on the same device. In the above example the 016 (octal) device address is the Sector Device Address of the drum unit and the function 3 indicates the Drum Busy condition line. Currently this is the only drum condition which can be sensed.

Drum Unit Device Addresses and Function Codes

Two I/O device addresses are assigned to the drum unit. The drum sector device address is 017 octal and the drum track device address is 016 octal. The names sector and track are appropriate for the manner in which these addresses are used in the drum single character write, the drum line read/write and the select drum track for CRT viewing operations. The drum read/write operations require a two word command sequence to be output to the drum controller from the 620i. The sector word is first issued to the sector device address and then the track word is issued to the track device address.

The track device address (016) is also used in a single word command sequence to select a drum track for CRT viewing at an Intrex display console.

The sector device address (017) is used as an input address to input the contents of the drum sector counter.

The track device address (016) is used as an input address to input the contents of the drum revolution counter.

To sense the Drum Unit BUSY condition either of the drum device addresses (016 or 017) may be used in the Sense instruction. For the BUSY condition the proper function to use is 3 and the 9 bit combined function-address may be either 0316 or 0317 (octal). No other condition may be sensed on the drum unit so 3 is the only function code used with the drum sense operation.

The 620i External Control (EXC) instruction is used to initiate certain drum unit control operations. The 3 bit function code used in the instruction determines which control operation is to be performed. The 6 bit device address also contained in the instruction to indicate which I/O device is to be selected may be either 016 or 017 (octal). Either device address may be used with the EXC instruction to select the drum unit with identical results. This interchangeability of the drum device addresses is true only for use in the EXC and SEN (Sense) instructions.

A list of the function codes used in EXC instructions for drum unit control operations follows.

Drum Control Operation	Function Code
Turn Off (Disable) Drum Interrupt Switch	3
Initialize Drum Unit (Puts the drum in a Ready condition)	5
Turn On (Enable) Drum Interrupt Switch	6
Initialize Revolution Counter (Zero contents)	7
Connect/Disconnect BIC Write*	0
Connect/Disconnect BIC Read*	1

* The BIC (Buffer Interlace Controller) feature is not instrumented in the Intrex Display System at this time; however, the drum controller does have hardware that is associated with the BIC function in case of system modification.

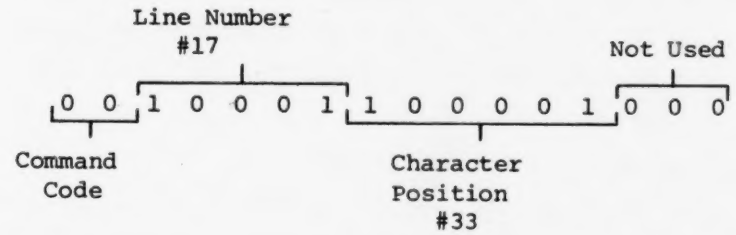
A discussion of these control operations follows below.

Write Single Drum Character Example-Command Format and 620i Program Segment

In the write single drum character command sequence the actual character code to be put into drum storage is contained in the track word

of the two word sequence. The command word formats are presented in an example below.

Sector Word for Write Single Drum Character



The 16 bit sector word contains the following information:

Command Code — the two most significant bits, 00 indicate the Write Single Character Drum Command.

Line Number — a 5 bit number indicating the drum storage line to be used; Line Number 17 in the example.

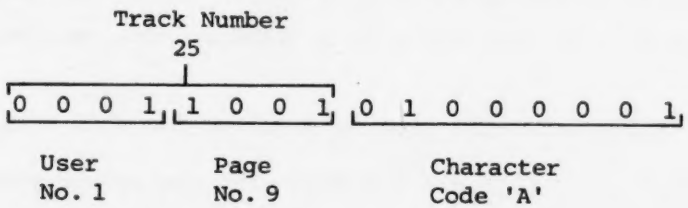
Character Position — a 6 bit number indicating the position within the selected drum storage line on which the character code is to be written; Character Position 33 in the example.

Sector Number — the 11 bit concatenation of the Line Number with the Character Position; 1121 (decimal) or 02141 (octal) in the example.

The low order 3 bits of the sector word are not used in the Write Single Character Command sequence.

The octal representation of the complete sector word for the above example is 021410.

Track Word for Write Single Drum Character



The 16 bit track word contains the following information:

Track Number — the 8 most significant bits indicate the track to be used in the operation; track 25 in the example.

User Number — the 4 most significant bits of the track number can be taken as a user or console number. Track numbers between 16 and 31 (inclusive) can be viewed at console number 1. User Number 1 is indicated in the example.

Page Number — the 4 least significant bits of the track number can be taken as a Page Number for the given user. A user's page numbers range from 0 to 15. The example indicates page number 9 is to be used for the write operation.

Character Code — the 8 least significant bits of the track word contain the ASCII code for the character which is to be written into drum storage. In the example the code represents an alphanumeric A whose octal code is 0101. The most significant bit of the 8 bit character code in ASCII is the parity bit. The drum control unit merely transfers the bits it receives and does not perform any parity checking. (For a further discussion of the parity bit contained in the character code within the Intrex Display System consult a section of this report given below.)

The octal representation of the complete track word for the above example is 014501.

Consider the 620i instructions required to command the drum controller to write a single character on the drum. First the drum must be sensed for a BUSY condition. When a NOT BUSY condition exists the 620i outputs a sector word to the sector device address and then a track word to the track device address. The sector word contains the command code for the write single character operation, a line number and character position to indicate where the character is to be written on the selected drum track. Both the character code to be written and the drum track to be selected are indicated in the track word.

A possible 620i program sequence for this example follows:

```
,LDA,SW          Load A register with the sector word
,LDB,TW          Load B register with the track word
,SEN,0316,*      Wait for Drum Unit NOT BUSY
,OAR,017         Output A register to Sector Device
,OBR,016         Output B register to Track Device
```

```
,JMP,CONT          Jump to continue program
SW,DATA,021410     Sector Word
TW,DATA,014501     Track Word
```

The above program sequence loads up the A register with the sector word and the B register with the track word. The SEN instruction causes the CPU to wait until the drum unit is not busy. Drum Busy is the condition which is being sensed and as long as the condition is true (i.e., the drum is busy) the 620i takes the address of its next instruction from the second word of the sense instruction. In this program the * indicates the address of the sense instruction itself. As long as the drum remains busy the CPU refetches the sense instruction from core and executes the test again. The program will continue in this single instruction loop until the drum unit becomes not busy or until a device interrupt breaks the normal program sequence.

When the drum becomes not busy (enters a Ready condition) the CPU then outputs the sector word to the Sector Device and the track word to the Track Device. The drum control unit then takes over the task of writing the character on the correct sector of the drum. All the needed information was supplied to the drum controller in the sector and track words. The amount of time that the drum write requires depends upon the physical position of the revolving drum. The maximum delay would be the time required for one drum revolution, 17 ms. However, the CPU only needs to issue the two command words and may then proceed on to execute other instructions. The drum controller does the waiting for the correct sector to come under the write head before completing the drum operation. Only if the program encounters another drum operation does the CPU need to wait for completion of the current drum operation. Again the Sense instruction can be used to test if the drum operation is over and the drum unit is Ready. The interrupt capability of the 620i may also be used to indicate when the drum unit is ready.

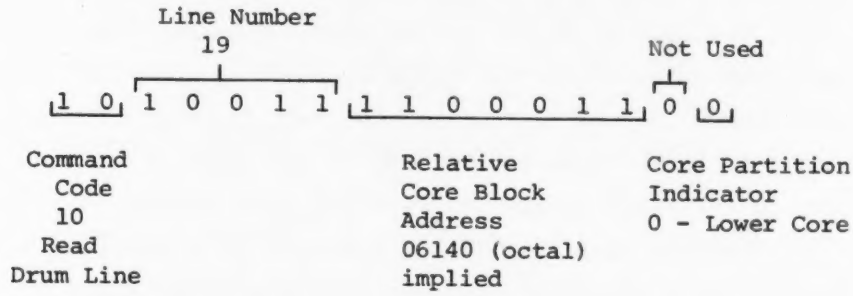
Read/Write Drum Line Example-Command Format and 620i Program Segment

A block of 32 words, each word 16 bits long, can be written from 620i core onto the drum or read from the drum into core by outputting a two word command sequence to the drum control unit.

The first word (the sector word) contains the command code identifying the command sequence as a read drum line or a write drum line operation, the drum line number indicating which line is to be transfered, and the core address indicating which block of memory locations are to be used. The only information contained in the second word of the command sequence is the drum track number.

The command word formats are presented in an example below.

Sector Word for Read/Write Drum Line


Line Number
19
Not Used
1 0 1 0 0 1 1 1 1 0 0 0 1 1 0 0
Command Code 10 Read Drum Line
Relative Core Block Address 06140 (octal) implied
Core Partition Indicator 0 - Lower Core

The 16 bit sector word contains the following information:

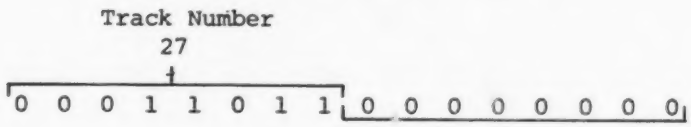
Command Code — 1 0 Read Drum Line, transfer 32 sixteen bit words from drum storage into core.

Line Number — 10011 binary, 19 decimal. Indicates the drum line to be read.

Relative Core Block Address — 1100011 binary, implies relative core address 06140 octal. The relative block address is obtained by adding 5 binary zeros to the low order of the given relative core block address, 110001100000. This address is still relative to the core partition and must be altered in accordance with the core partition indicator value.

Core Partition Indicator — 0, indicates that the lower partition of core is to be used. This means that the relative block address is taken relative to the lower partition boundary at 00000. Thus the absolute core address to be used as the starting address for the block transfer of 32 words is 0110001100000 binary or 06140 octal.

Track word for Read/Write Drum Line



The only information contained in the track word for the read/write drum line command sequence is the number of the drum track to be used in the operation. The high order 8 bits of the track word contain the track number which is 27 decimal in the above example.

A possible 620i program segment to read a drum line follows.

```
    ,LDA,SW
    ,LDB,TW
    ,SEN,0316,*
    ,OAR,017
    ,OBR,016
    ,JMP,CONT
SW,DATA,0123614
TW,DATA,015400
```

The detailed description of this program segment is essentially the same as that given in the previous section for the Write Single Character operation. The command words output to the drum device addresses differ in content for the two examples and so the drum operations themselves will be different even though the 620i program structure is the same. Timing considerations for the Read/Write Drum Line operations are discussed in the following section.

Timing Considerations for Read/Write Drum Line Operations

In read/write drum line operations the 620i CPU starts the I/O block transfer between core and drum storage by outputting the drum command sequence to the sector and track device addresses. The CPU can continue on with the execution of other instructions while the drum controller continues the block transfer of data using the mechanism of direct memory access and cycle stealing from CPU control by the 620i

trap control feature. The trap feature allows direct transfer of data between core memory and the 620i I/O bus under the control of the I/O device.

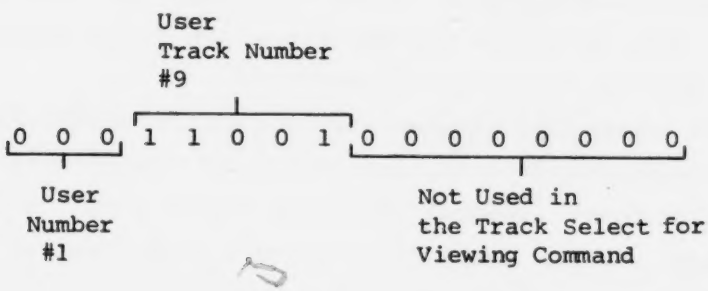
The drum controller initiates trap requests which are acknowledged by the CPU within a fixed time limit. This limiting time depends upon conditions within the CPU and upon the priority assigned to I/O devices having trap and interrupt capability. The drum controller has been assigned the highest I/O priority in the Intrex Display System because it has the highest data transfer rate of the system I/O devices. The drum block transfer operation reads or writes at the rate of 120,480 characters per second. The maximum delay of a trap request acknowledgement and servicing appears to be something less than 7.2 µs (Perhaps 900 nanoseconds less). The drum controller appears to require service in 7.47µs.

In the Intrex Display System the trap mechanism allows the drum controller to transfer data by block to or from adjacent drum sectors as they pass beneath the read/write heads. A block of 32 sixteen bit words are transfered in 0.53 ms after the first sector begins to pass under the read/write heads. It takes 16.6µs for a drum word which consists of two characters to pass beneath the heads. During the 16.6µs the controller cycle stealing consumes approximately 3.6µs leaving 13.0µs for CPU operation between stolen cycles. Thus during the 0.53ms of block transfer about 28% of CPU time is stolen so that CPU operations proceed at approximately 72% of normal speed during the transfer.

Select Drum Track for CRT Viewing — Example Command Format and 620i Program Segment

To assign a track to supply CRT refresh information to the Intrex console the 620i program must first wait for the drum ready condition and then output the appropriate track word to the track device address 016 octal.

Our Example of the Track Word Format is:



A possible 620i code sequence to select a drum track for viewing on the Intrex CRT display console follows:

```
,LDB,TW
,SEN,0316,*
,OBR,016
,JMP,CONT
TW,DATA,014400
```

The drum control unit recognizes that outputting a command to the track word address 016 (octal) without preceeding it by an output to the sector word address is a command to select that track for viewing at the indicated console.

In the example since user console #1 was selected only the tracks from 16 (decimal) to 31 (decimal) inclusive could be assigned to that console. In the example track #27 (decimal) was selected and it corresponds to the user's page number 9. The user's page numbers can range from 0 to 15 (decimal).

Read Sector Counter — 620i Program Sequence

The contents of the drum sector counter are input by execution of a 620i input instruction with the sector device address 017 octal. It is necessary to check for a Drum NOT BUSY condition before reading the sector counter, otherwise a drum operation started previously and not yet complete might be disrupted and yield undesirable results. A 620i program sequence to read the drum sector counter might be as follows.

```
,SEN,0316,*
,CIA,017
```

The sense instruction causes the CPU to wait for the drum not busy condition before execution of the Clear and Input to A Register instruction. The CIA leaves the contents of the sector counter divided by 2 right justified in the A register.

The Intrex 620i programs read the sector counter as part of an algorithm used to reduce blanking of the CRT display during drum read/write operations. Reading the sector counter makes it possible to issue drum commands just prior to the passage of the desired sector beneath the read/write heads.

Read Revolution Counter 620i Program Sequence

The contents of the drum revolution counter are input to the 620i by execution of an input instruction with the track device address 016 (octal). It is not necessary to check for the drum ready condition before reading the revolution counter. Any 620i input instruction may be used, such as:

```
,CIA,016
```

The Clear and Input to A Register instruction clears the A register and leaves the contents of the revolution counter in the A register. The drum revolution counter may be initialized, its contents set to zero, by use of the 620i External Control Instruction (EXC) with a device address of either 016 or 017 and a function code of 7 as follows:

```
,EXC,0716     or     ,EXC,0717
```

The System Reset Switch on the 620i control panel also sets the revolution counter to zero.

620i I/O Device Interrupt Capability

The interrupt feature of the 620i allows an I/O device to interrupt the normal execution of a 620i program sequence. When an I/O device attempts to issue an interrupt request, if the proper conditions exist in the I/O interface and in the 620i CPU, then the interrupt request is made and the CPU acknowledges the request, accepts an interrupt address from the device and then takes its next instruction from

the core location indicated by the interrupting device. Various I/O devices may force the CPU to execute instructions at their particular interrupt addresses. Depending upon which 620i instruction is stored at the interrupt address used, the CPU may continue in its normal program sequence or may enter a special interrupt handling subroutine.

The instruction obtained from the interrupt location may be any 620i instruction except I/O instructions. If the instruction is not an I/O or a jump instruction, the CPU executes that instruction and then continues the program sequence it was executing prior to the interrupt. Any jump instruction would cause a corresponding program branch. If the instruction was the JMPM (Jump and Mark), the execution of this instruction during the interrupt cycle causes the Master Interrupt Switch to be turned off disabling all further interrupts until the 620i program turns the Master Interrupt Switch back on. The JMPM may be used to cause the CPU to enter an interrupt servicing subroutine, saving the location of the next instruction which would have been executed if the 620i program had not been interrupted by the I/O device. Later the interrupt subroutine may return to the original program sequence, probably after re-enabling the Master Interrupt Switch.

In order that 620i two word instructions may be executed properly at the interrupt core address, that interrupt address should be an even numbered core address. This is a hardware requirement and all interrupt addresses generated by Intrex Display System I/O devices meet that requirement.

The Master Interrupt Enable Switch may be turned on or off by 620i program execution of an EXC (External Control) instruction containing the correct 6 bit device address and 3 bit function code. Each I/O device with interrupt capability has associated with it a Device Interrupt Switch which can be turned on or off by EXC instructions with appropriate device addresses and function codes. For a particular interrupt to occur both the Master Enable and the particular device enable switches must be on (Enabled).

I/O Device Interrupt and Trap Priorities

When the instruction at the interrupt address is a JMPM (Jump and Mark) the execution of this instruction turns off the Master

Interrupt Enable Switch thereby inhibiting the initiation of other interrupt requests. When the 620i program re-enables the Master Interrupt Switch then the I/O device with the highest interrupt priority and a pending interrupt request condition is allowed to initiate the next interrupt request. Thus high priority I/O devices requiring interrupt service can hold off low priority devices also requiring interrupt service. The priority hierarchy is hard-wired into the system as are the interrupt core addresses, the device addresses and the device function codes.

Trap requests are not delayed by the Master Interrupt Switch being off. Trap requests may be initiated, acknowledged and data transfered using the DMA (Direct Memory Access) hardware even while the 620i program is executing an interrupt subroutine. The 620i hardware design insures that at least one instruction is executed between each trapping sequence. Thus a program such as the interrupt subroutine cannot be completely locked out by consecutive trap sequences of an I/O block transfer.

The same priority hierarchy applies to trap requests as to interrupt requests. Thus a high priority device requiring trap service can delay a lower priority device requiring either trap or interrupt service. The trapping sequence requires 2.7 μs for completion after the CPU acknowledgement of the trap request. Each 16 bit word transfer of a block data transfer requires a full trap sequence.

Drum Unit Priority Interrupt

The Drum Control Unit may be used in conjunction with the 620i priority interrupt hardware to indicate a Drum Ready condition. A Drum Ready Interrupt can occur if the Ready interrupt pending condition exists in the drum controller, the master interrupt switch is enabled in the I/O interface, the Drum Interrupt Switch is enabled in the drum interface and the CPU has just completed the execution of an instruction.

The drum unit is assigned the highest interrupt priority among the Intrex Display System I/O devices. The device priority scheme does not allow simultaneous interrupt requests from different I/O devices. If various devices require interrupt or trap attention at the same time, the device with the highest priority issues the interrupt or trap request when conditions permit it to do so. After the CPU

acknowledges a drum interrupt request the drum controller furnishes the CPU with the drum interrupt core address (010 octal) and the CPU executes its next instruction from that address.

The detailed circumstances under which the Drum Interrupt occurs is of some importance to the 620i programmer. After the drum unit has completed a drum read or write or a select track for CRT viewing operation, a drum interrupt pending switch is set. The setting of the interrupt pending switch indicates that the Drum Busy condition caused by one of the above drum operations has been removed by completion of that operation. If the Master Interrupt Switch and the Drum Interrupt Switch are both enabled and the drum interrupt pending switch is set, the drum unit may initiate a CPU interrupt sequence. Completion of that interrupt sequence turns off the interrupt pending switch.

The interrupt pending switch may be turned off in two other ways. The Drum initialize operation resulting from the execution of the appropriate 620i EXC instruction and the manual System Reset operation initiated from the 620i console System Reset button both cause the interrupt pending switch to be turned off. (They cause other operations also.)

The conditions under which the interrupt pending switch are turned off are of importance if the 620i program is to properly handle the Drum Ready interrupt when it occurs. At the exact moment when the interrupt pending switch is turned on the drum unit is in a Ready condition. However,the occurrence of the Drum Ready Interrupt can be delayed by the 620i program disabling of either the Master Interrupt of Drum Interrupt Switches. If while the Drum Ready interrupt is delayed the 620i program uses a Sense instruction to detect the Drum Ready condition and then proceeds to begin a new Drum operation which causes the Drum Unit to re-enter a Busy state, at that time the interrupt pending switch falsely indicates a Drum Ready condition while the Drum is actually in a Busy state. If the 620i program then enables the Interrupt Switches before the Drum operation is complete,the Drum Ready Interrupt will occur.

It is up to the 620i programmer to either avoid the above circumstances or else make certain that the results are not

misinterpreted. Special care must be exercised when using both Sense instructions and interrupt signaling to detect the Drum Ready condition in the same program.

Enabling/Disabling Drum Interrupts

The External Control (EXC) instruction of the 620i must be executed to turn the Drum Interrupt Switch On (to Enable). Either of the following instructions may be used:

,EXC,0616 or ,EXC,0617

With the Drum Interrupt Switch On (Enabled) the drum controller may initiate interrupt requests when the proper conditions prevail in the CPU, the I/O interface and the drum unit itself.

To turn the Drum Interrupt Switch Off either of the following instructions may be executed:

,EXC,0316 or ,EXC,0317

With the Drum Interrupt Switch Off (disabled) the drum controller interrupt requests are not initiated even though a drum Ready interrupt pending condition exists in the drum controller. The interrupt pending condition persists in the controller until either it can initiate a CPU interrupt sequence and that sequence is completed, or until the interrupt pending condition is removed by either a programmed drum initialization or a manual system reset. Thus the Drum Interrupt Switch makes it possible for the 620i program to delay Drum Interrupt Requests until the program is ready to handle them.

Master Interrupt Enable/Disable Switch

The Master Interrupt Switch can be enabled (turned on) or disabled (turned off) by the execution of 620i EXC instructions with the appropriate device address and function codes.

To enable the Master Interrupt Switch the following instruction must be used:

,EXC,0270

To disable the Master Interrupt Switch the following instruction may be used:

,EXC,0470

The 6-bit device address (070 octal) in the above instructions are used in the Intrex Display System for the CTSS Data Phone Interface. However, when the device address 070 (octal) is used with the function codes 2 or 4 in an EXC instruction, the Master Interrupt Enable/Disable Switch is operated.

The Master Interrupt Switch is also disabled (turned off) by use of the manual System Reset Button on the 620i console.

Unless the Master Interrupt Switch is on (enabled) none of the Intrex Display System I/O devices can initiate a 620i CPU interrupt sequence.

Initialize Drum Unit and 620i Manual System Reset

The 620i EXC instruction may be used to initialize the drum unit as follows:

,EXC,0516 or ,EXC,0517

The execution of either instruction causes the drum control unit to enter a Ready condition and to turn the Ready interrupt pending switch off. The initialize command also turns the Drum Interrupt Switch off (disables the interrupt).

Manual operation of the System Reset button on the control panel of the 620i also initializes the drum unit. Use of the System Reset button initializes all I/O devices in a manner appropriate for each device and disables the Master Interrupt Enable/Disable Switch. It also resets the Drum Revolution Counter to zero.

Note on Drum Interrupt Terminology Regarding Switch and Mask

In the ESL-TM-410,the Augmented-Catalog Console for Project Intrex (Part II) by Donald R. Haring, December, 1969, a special terminology for the Drum Interrupt Switch is used. In that publication on page 49 Turn on Drum MASK Interrupt corresponds to Turn off Drum Interrupt Switch as used in this report while Turn off Drum Mask Interrupt corresponds to Turn on Drum Interrupt Switch.

The terminology used in this report is consistent with that used by Don Haring in TM-410 for I/O devices other than the Drum. Some 620i manuals refer to interrupt masks; however, the terminology used there is not clear nor fully defined.

Note on Character Codes Within the Intrex Display System

In the Intrex Display System eight data bits are allotted for character codes. In general a 7 bit ASCII code is observed while the 8th bit, the high order bit, is sometimes treated as a parity bit and sometimes not. The drum controller handles this 8th bit just like the other 7, as a data bit which it puts onto or takes off of the E-bus. The CRT display circuitry only looks at the 7 data bits. The codes mentioned in this report are given as 7 bit codes with the 8th bit taken as zero. Thus DEL becomes 177 in octal.

Summary of Drum Controller/620i Addresses and Codes

Drum Device Addresses (6 bits)

Sector Device 017 (octal)

Track Device 016 (octal)

External Control Function Codes (3 bits) (Either 016 or 017 octal device address may be used)	Code
Turn Off Drum Interrupt Switch (Disable)	3
Initialize Drum Unit	5
Turn On Drum Interrupt Switch (Enable)	6
Zero Revolution Counter	7
Connect/Disconnect BIC Write*	0
Connect/Disconnect BIC Read*	1

* The BIC option is not instrumented in the Intrex Display System

Sense Function Codes (3 bits)

(Either 016 or 017 octal device address may be used)

Drum Busy 3

Jump occurs on drum condition of Busy
(The Busy condition corresponds to the Not Ready condition)

Drum Ready Interrupt

Interrupt Core Address 010 (octal)

I/O Data Transfer

Track Device 016 (octal)

16 bits from 620i to E-bus 00-15 Track Data for drum controller

Sector Device 017 (octal)

16 bits from 620i to E-bus 00-15 Sector Data for drum controller

10 bits to 620i from E-bus 00-09 Sector Counter Contents

Drum Command Code (Used in Sector Word)

Code	Drum Operation
00	Write Single Character (8 bits)
01	Write Drum Line (64 characters)
10	Read Drum Line (64 characters)
11	BIC Control (Buffer Interlace Controller)

Collection of Drum/CRT Terms, Parameters and Specifications

Intrex Display System (ESL designed and built)

Buffer-Controller (ESL designed and built)

Vermont Research C747 Magnetic Drum Unit

Drum Control Unit (also called Drum Controller)
(ESL designed and built)

Varian Data Machines 620i/8K core

DMA (Direct Memory Access) - 620i feature

Current Drum System

Storage Capacity

655,360 bits

65,536 sectors (10 bits/sector)

65,536 characters (8 data bits each)

32 tracks

2048 ten-bit sectors per track

2048 eight-bit characters characters per track

1024 sixteen-bit words per track

Drum Speed - 3528 rpm

Drum Revolution time - 17 ms

Max Drum Read/Write Latency Time - 17 ms

Average Drum Read/Write Latency Time - 8.5 ms

Data Transfer Rate	120,480 sectors (characters) per second
Data Transfer Time	8.3 µs per sector (character) 16.6 µs per 16-bit word
Block Transfer Time	0.53 ms per 64-character block (32 words)

CRT Display

One drum track refreshes one CRT page display.

Each Intrex console has access to a maximum of 16 drum tracks for direct viewing (a hardware limitation).

1736 visible character positions per display page

31 lines of 56 character positions per page

Page synchronization — ASCII character pair in adjacent sectors:

DEL — 0177 followed by SYN — 026

(Given in 7-bit code in octal — the 8th bit is not observed by the console and may be either zero or one.)

Drum Sector Address — 11 bits; a 5-bit line number and a 6-bit character position

Track Number — 8 bits; a 4-bit user number and a 4-bit page number.

Drum Sector Counter — mod 2048 divided by 2 on input to 620i

Drum Revolution Counter — mod 2^{16} (65,536 decimal)

Range as clock 0 to 1,114,11 seconds (18.56 minutes)

Time increments of 17 ms.

APPENDIX C

DRUM FILE OF PROTOTYPE DISPLAY LABEL LINES

In the current system drum track 12 is employed to store the prototype of display label lines. The value of the user parameter MODE indicates the drum line number for the corresponding label line.

For MODE=1, which corresponds to the Catalog Mode, the label prototype is retained in core storage to speed label writing in that Mode. The drum line #1 label is not used and the core label for Catalog mode follows:

```
 ERASE   NOTE    NCLEAR                  NVIEW       PAGE
```

The contents of the Label Track (LBTK) follows.

```
 CATG    TXTAC   *START*   LOCAL
           You may now exercise the catalog
*text access*                  FIRST   LAST          CLEAR
ERASE   CLEAR  VIEW   *edit*  ERLN   EROP   SVLN    RTLN
*local mode*    SHP    EDIT
*LOCAL MODE*  You may now type at will.
        SHP     TRTK   EDIT   DSPS   CNDS   PSOD    GSFD
*gsfd*  Enter two parameters from the keyboard
*psod*  Enter two parameters from the keyboard
*dsl*   Enter three parameters from the keyboard
*trtk*  Enter three parameters from the keyboard
*csl*   Enter three parameters from the keyboard
 ATRK
ERASE  CLEAR  RETURN  *view*  ERLN   EROP   SVLN    RTLN
```

APPENDIX D

INDEX OF FUNCTION CODE TABLES AND INDEX OF ADDRESS TABLES

Location	Source Listing	Argument Value	Comment
03000	LFNT,PZE,FT1	(016560)	Points to first item in Function Code Table Index
			Mode
016560	FT1,PZE,FNT1	(03005)	Catalog, Text Access
016561	,PZE,FNT2	(010705)	Edit
016562	,PZE,FNT3	(010705)	View
016563	,PZE,FNT4	(016522)	Demonstration
016564	,PZE,FNT5	(013700)	Source Handling
03026	LAT,PZE,AT1	(016570)	Points to first item in Address Table Index
			Mode
016570	AT1,PZE,FJAT	(03032)	Catalog, Text Access
	,PZE,SJAT	(010732)	Edit
	,PZE,TJAT	(016601)	View
	,PZE,AD4	(016525)	Demonstration
	,PZE,AD5	(013711)	Source Handling

APPENDIX E

FUNCTION ROUTINE LIST WITH CALLING MODES

The following Function Routines are called from FSDE (Function Switch Decode) in response to use of the Function and Programmable Switches at the BRISC terminals.

This Appendix gives the function routine name, its absolute core location and the program modes from which the function may be called. The following abbreviations are used to indicate the program modes:

C	Catalog
T	Text Access
E	Edit
S	Source Handling
V	View
D	Demonstration

The Function Routines are listed in the following groups:

Display Page Controls
Mode Control
Text Display Controls
Note or Save Page Control
Edit Controls
Source Handling Functions

Name		Calling Modes	Location
Display Page Controls			
GNW	Get New Page	C,T	016300
SFP	Select First Page	C,T,E,S	01605
SNP	Select Next Page	C,T,E,S	01614
SPP	Select Previous Page	C,T,E,S	01641
CERS	Call Erase	C,T,E,S,V	01665

APPENDIX E (Cont'd)

Name		Calling Modes	Location
Mode Control			
NCAT	New Catalog	C,T	016420
NLOC	New Local	C,T,E	016400
ETA	Enter Text Access	C,T	03100
CURM	Cursor Mode	C,T,E,S,V	01042
CFVE	Catalog From View	E,V,D	016432
EDIT	Edit	C,T,E,S	010300
VMDE	View Mode	C,T,E	016466
FPFV	First Page From View	V,D	016443
VSMP	View Sample	V	016510
RFY	Return From View	V	016454
Text Display Controls			
FPR	First Page Routine	C,T	011166
LPR	Last Page Routine	C,T	011410
SPR	Same Page Routine	C,T	011205
NPR	Next Page Routine	C,T	011000
PPR	Previous Page Routine	C,T	011006
EMG	Enter Magnify	C,T	011232
CR	Copy Routine	C,T	011500
ZT	Zero Text	C,T	011400
Note or Save Page Controls			
CSVP	Clear Save Page	C,T,E,V	010334
NTLN	Note Line	C,T	016530
Edit Controls			
EL	Erase Line	E,V	010307
EROP	Erase Rest of Page	E,V	010500
RTL	Return Line	E,V	010404
SVL	Save Line	E,V	010342

APPENDIX E (Cont'd)

Name		Calling Modes	Location
Source Handling Functions			
EDSL	Enter Display Source Lines	S	013033
ECSL	Enter Condense Source Lines	S	013044
EPSD	Enter Put Source On Drum	S	013022
EGSD	Enter Get Source From Drum	S	013011
ETTK	Enter Transfer Tracks	S	013000
CSH	Call Source Handling	S	013660
EATK	Enter Advance Track	S	013613

APPENDIX F

FUNCTION, PROGRAMMABLE AND TEXT ACCESS SWITCH CODES

Below is listed the eight bit function codes generated by the BRISC function, programmable and text access switches. The code is a three out of eight code and the code polarity given is that which is input into the 620i general registers such as the A register.

The programmable switches are numbered from left to right as viewed by the user. The function switches are numbered from left to right as they appear on BRISC 1. On BRISC 2 the physical arrangement is different. In order to clearify this difference the current use of the function and text access switches are given with the code and switch number.

Function Switch #	Function Code	Use
1	340	New Page
2	320	First Page
3	310	Next Page
4	304	Previous Page
5	302	Edit
6	301	Text Access
7	260	Catalog
8	250	Cursor

Programmable Switch #	Function Code	Use in Catalog/ Text Mode
1	244	Erase
2	242	Note
3	241	Note Clear
4	230	Note View
5	224	First Text Page
6	222	Last Text Page
7	221	Copy Text
8	160	Zero Text

APPENDIX F (Cont'd)

Text Access Switch #	Function Code	Use
1	130	First Page
2	033	Next Page
3	015	Previous Page
4	016	Same Page
5	052	Last Page
		Sector Numbers
6	142	1
7	242	2
8	302	3
9	144	4
10	244	5
11	304	6
12	150	7
13	250	8
14	310	9
15*		Magnify

* No longer generated by any existing switch.

APPENDIX G

SYSTEM MESSAGE TRACK CONTENTS

The contents of the current system Message Track (MTK) is listed below. Less than half of the messages are used in the current operating system. The others are left over from previous versions of the system. Those in current use are marked on the left with an *.

* MESSAGE LINE
* TEXT RETRIEVAL SYSTEM READY PAGE
* TEXT SYSTEM ERROR 2 PAGE
* REQUESTED FICHE NOT FOUND PAGE
* TEXT SYSTEM ERROR 4 PAGE
* TEXT SYSTEM ERROR CENTRAL STATION TIMER RUNOUT PAGE
* TEXT SYSTEM HANDSHAKE PAGE
* TEXT SYSTEM MALFUNCTION ERROR 7
* LOCAL BUFFER OVERFLOW OUTPUT LOSS OCCURRED
* *ERASE* PRESS ERASE TO CONTINUE — THIS IS DIALOG PAGE
T.A.WORD # OF ENTRIES IN QUEUE TOP BOTTOM
MANIFY OPTION; ENTER SECTOR # AS SINGLE DIGIT 1 TO 9
SECTOR # XMIT MODE: TYPE C FOR COPY OR D FOR DISPLAY
OCTAL SECTOR # TYPE 3 CHARACTERS FOR QUEUE COLUMN #
X SECTOR PREVIOUS TEXT REQUEST NOT COMPLETE ROW #
Y SECTOR NO DISPLAY REQUEST IN QUEUE
* FICHE FRAME DISPLAYED IS THE FIRST PAGE PAGE
* FICHE FRAME DISPLAYED IS THE LAST PAGE PAGE
FRAME # FICHE #
OF PAGES
CURRENT PAGE
HALT COMMAND ENCOUNTERED. PRESS RUN TO RECYCLE.
INPUT COMMAND NOT RECOGNIZED.
TEXT ACCESS MODE
DISPLAY OPTION
COPY OPTION
MAGNIFY OPTION
QUEUE IS FULL
COMMAND NOT RECOGNIZED
QUEUE IS EMPTY
ERASE

APPENDIX H

SOURCE HANDLING PROGRAMS —
DESCRIPTION OF THEIR USE

H.1 USE OF THE SOURCE HANDLING ROUTINES

The following functions are available in Source Handling Mode:

TRTK Transfer Tracks:

Transfers the contents of n consecutive tracks to n consecutive tracks. Requires the following three parameters from the user; First From (Source) Track, First To (Destination) Track and number n, of tracks to be transferred.

DSL Display Source Lines:

Takes source statements from consecutive packed drum tracks, unpacks and displays them, one statement per line (unpacked) on consecutive drum lines. Requires the following three parameters from the user; First from Track, First To Track and Last To Track plus one.

CSL Condense Source Lines:

Takes source statements from unpacked or partially packed consecutive drum tracks and packs them onto consecutive tracks. Requires the following three parameters from the user: First From (Source) Track, First To (Destination) Track and a Last From Track plus one.

PSOD Put Source on Drum:

Reads source statements from the paper tape reader and packs them onto the given drum track. Requires the following two parameters from the user: First To (Destination) Track and the Last To Track plus one.

GSFD Get Source From Drum:

Takes source statements from the given drum track and outputs them to the paper tape punch. Requires the following two parameters from the user: First From (Source) Track and the Last From Track plus one.

The plus one on the ending track is a simple safety measure. It causes the transfer to terminate one track short of the typed parameter.

SDI

H.2 ENTRY INTO SOURCE HANDLING MODE

In the current system a BRISC terminal can be placed into Source Handling Mode only by starting the 620i at either S1 for BRISC 1 or at S2 for BRISC 2. The program sequence from either S1 or S2 sets the mode for the given console and flushes the Keyboard String for that console before branching to CKC (Check Consoles) to enter the Monitor Loop. Once in the Monitor Loop the BRISC consoles may be operated in the usual manner.

The S1 and S2 program sequences alter only MODE and TBIX for the corresponding console. The condition of the interrupt switches is not altered nor the user parameters of the other BRISC. The user may enter Edit Mode from Source Mode at the BRISC, however, he cannot reenter Source Mode without doing so at the 620i. This is to prevent normal Intrex users from gaining access to the Source Programs from the BRISC terminals. The Edit Mode Address Table can be modified at the 620i so that the *edit* programmable switch calls SHP (Source Handling Program) instead of EDIT. This change can be made for a console session and afterwards the table may be restored.

Entry Points S1 and S2 are used to place BRISC 1 and BRISC 2 respectively in Source Handling Mode. Only one BRISC can be in Source Mode at a given time. The other BRISC may be operating in the non-SHP modes, however, due to the extensive drum operations used in Source Mode, if a second console operates simultaneously in Catalog Mode, some CTSS data may be lost.

H.3 FUNCTION PARAMETERS AND FUNCTION EXECUTION

The Source Handling Function programs require parameters from the BRISC user. The user presses the desired programmable switch and the resulting label line identifies the function selected and instructs the user to enter either 2 or 3 parameters from the keyboard. The number and nature of the parameters depend on the function selected.

The parameters should be typed on a new line. The usual procedure is to enter the Source Handling Mode, select and Erase the desired working display page, press the function switch and then type the number of parameters requested ending with a CR (Carriage Return). The carriage return indicates the end of the parameter string and starts the selected function routine.

PSOD (Put Source On Drum) and GSFD (Get Source From Drum) both require two parameters, a starting track number and an ending track number plus one. Drum track numbers are typed as decimal numbers representing the absolute drum track number which ranges from 0 to 127. BRISC #1 (located in ESL) display tracks are numbered from 96 to 111, and those of BRISC #2 (in Barker Library) are numbered from 112 to 127. System Tracks are numbered from 0 to 95.

DSL (Display Source Line) and CSL (Condense Source Lines) both require three parameters. DSL requires a From (Source) Track number followed by a To (Destination) Track number followed by the Last To Track Number plus one. CSL requires a From Track Number, a To Track Number and a Last From Track Number plus one. The third parameter in both cases is for the unpacked data which usually occupies more tracks than the packed data.

As the function is being executed and as the processing of each unpacked page is completed, the ATRK (Advance Track) label appears. The user must press ATRK before the next unpacked page will be processed. If the user starts the operation by typing the parameters on the First To page, then he sees the page fill with text. For DSL when the ATRK appears he may press Next Page to page forward, Erase to clear the page and then ATRK to continue the operation. Using this technique reduces the chances of overwriting pages of needed data. For CSL four or five pages of unpacked data will fit on the To page and when ATRK appears and there is still room on the To page, the user only needs to press ATRK to continue.

The To Track parameter determines the destination of the data and the operation gives a pause when the bottom of the unpacked page is reached. Pressing ATRK causes the next page of unpacked data to be processed. The use of Next Page and Erase is merely a means of observing the action.

After the last To Track has been filled and the ATRK label appears, if the user presses ATRK again, the Source Handling Program is reentered giving a new set of labels. During the execution of the function, the user may intervene at any ATRK pause to terminate the operation. He does this by pressing any of the Source Handling buttons other than ATRK. Those buttons are active even though they may be unlabeled. The safest action is to press the SHP button to the right of the ATRK.

The user should keep in mind that while these Source Handling programs are fully debugged and have been extensively used by the author who designed and programmed them, they do not have any file protection mechanism. If the user supplies incorrect parameters or if he does not follow stated procedures, then system drum files may be destroyed and even the core image could be corrupted.

One point to note is that whenever one of the source functions has been entered and the monitor program is waiting for the function parameters to be typed, the receipt of a carriage return triggers the execution of the function routine. The parameter values are taken from the left-most end of the Keyboard String. The user must not be careless about entering a source function when he has partially filled the Keyboard String by some previous Keyboard action. Use of either the @ character to clear the line or the Erase to clear the page will also clear the Keyboard String. Typing the @ before typing the function parameters is a desirable procedure.

H.4 EXAMPLES OF SOURCE HANDLING

Suppose the user of BRISC #1 has source statements packed on tracks 102 through 104 and he wishes to unpack and display that source file on tracks 96 through 100. He then may select track 96 (his first display page), erase the page, press DSL and type the following parameter line left-justified on the top line of track 96.

102 96 101(CR)

Source statements are taken from packed drum tracks starting at 102 and placed, unpacked (one statement per line) on consecutive drum tracks beginning at 96. When track 96 is full the ATRK label will appear and the user should select track 97, erase it, and press ATRK. Then track 97 will fill and the procedure is repeated until track 100 has been filled. Then pressing ATRK will cause the SHP label to reappear and the operation is complete. As many consecutive input tracks starting at 102 are taken as are needed to fill tracks 96 through 100. The operation may be terminated at the end of any input track by pressing the SHP button, or any of the Source Handling function buttons (which are active even though their labels may not be visible) instead of pressing ATRK to continue it.

As a second example assume that TRTK (Transfer Tracks) has been pressed and the following parameter string has been typed left-justified on a clean line.

112 123 3(CR)

The contents of tracks 112, 113 and 114 will be transferred to tracks 123, 124 and 125 respectively. The CR actuates the transfer.

When the drum operations are being performed the display screen may blank for an alarming period of time. This is because the Source Function Routines do not employ CKSW (Check Sector Word) to locate drum sectors before issuing read/write drum commands as do the Monitor Routines.

Caution must be taken in using TRTK because no file protection is instrumented in the current system and any drum track can be overwritten. Care should be taken to get a new line (preferably a blank page) before pressing TRTK and typing the command parameters. It is very easy to write garbage on all or some portion of the 128 drum tracks.

H.5 SOURCE FILES FOR ASSEMBLY PROGRAM INPUT

The Source Handling programs are most useful in generating source files on drum tracks to be assembled by the Modified Assembly Program. New files may be generated by typing source statements at the BRISC terminal on the user display pages. The easiest technique is to type the Source statements in a semi-packed format on the display page. This is done in Cursor Mode, typing as many complete statements per line as will fit on the screen. Statements are separated by a LF (Line Feed) and the Cursor Controls Up and Right are used to position the Cursor for the next statement. Before beginning a page it should be Erased. This places null characters (all bits zero) on the track. The @ and # characters may be used to null a line or a character position. Null characters and lines may be left between statements. They are ignored by the assemble program. Nulls are edited out by the CSL routine but are transferred by both the PSOD and GSFD functions.

A parameter in the Modified Assembly Program is currently set to begin the input source file at user display page 1 of BRISC terminal 1. The source file occupies consecutive drum tracks and is terminated by

the,END, statement. No file labels or headers are processed. Source statement comments are used to identify visually the source text.

Leaving the Source Files partially packed makes corrections and modifications convenient after the execution of an assembly. The Edit features of the BRISC may be used as well as the Source Handling Functions to alter the Source Files for re-assembly. The only real use of CSL (Condense Source Lines) is to conserve space. There are only 16 display pages for viewing the Source Files at a BRISC so it is convenient to limit the Source File to those tracks. Other arrangements are possible but less convenient.

TRTK (Transfer Tracks) may be used to move Source Files onto any drum track. This includes system tracks and other BRISC display tracks; so the Source Files may be maintained on the drum. However, due to the perishability of drum files,GSFD (Get Source From Drum) and PSOD (Put Source On Drum) are available (and should be used) to output the Source File on paper tape and to read them back onto drum tracks. None of the Source Programs perform checking or verification. Of course assembly is the ultimate verification.

Use of PSOD and GSFD entails some special user procedures. When PSOD is to be used the paper tape Source File must be mounted with the first frame at the read position on the ASR-33B before the function is called. The TTY must be ON-LINE. The function routine will start the reader. If only one track is called for by the PSOD parameters, and if the Source tape has sufficient data to fill the track, then the ATRK label will appear at the BRISC and the user may terminate the execution of the function. However, the tape reader will continue to feed paper tape until the physical end of the tape is reached.

If the user has called for more than one track of paper tape input and if the tape is long enough, then the user must rapidly press ATRK when it appears on the label line, to advance to the next track. Some data may be lost and this operation is not usually very successful. (The difficulty arises because it is not possible to turn the reader off at a given input frame so that it can be restarted at the next frame.) Instead, paper tape Source files are usually generated containing the contents of only a single drum track. Use of elaborate buffering and data recovery techniques to correct the problem was not considered necessary for this application.

If the input paper tape is not positioned correctly when starting, or if the reader misreads the first character, then the Source File must be corrected on the displayed drum file by overtyping. Nulls, or LF (Line Feeds), may be typed to blank nonsense characters between statements. The Delete character may be used within the statement to remove an unwanted character. If the first line of the file is used for a comment identifying the file, then it is a simple matter to correct that line.

If the input file is a partially packed file and if the first character is not positioned properly, then the whole display may be shifted and some of the source text will not be visible on the screen. The best procedure is to restart the function until the lines are properly displayed. If the input is a fully packed file, then parts of the source are never visible since they occupy drum positions corresponding to the display retrace (8 characters per line). In this case the initial text line may be corrected and it is not necessary to restart the function.

If the input tape does not contain enough data to fill the drum track, the user may type additional input from the TTY. Use of the Repeat key is useful to flush any unfinished lines from the input line buffer. Since the user must return to the 620i site to use the TTY, he may wish simply to restart the program at S1 or S2 or he may type enough data to fill the track and terminate the function.

When generating paper tape source files with GSFD the function routine does turn the punch off when the last frame has been output. The function routine precedes the source text with a leader; however, it does not produce a trailer at the end. When the output is complete the ATRK label will appear and the user may go to the 620i site, place the TTY in OFF-LINE and press the HERE IS key once or twice to produce a trailer of nulls on the tape. He then tears the tape end and removes it from TTY. It must be marked manually for visual identification. The punch is turned off by holding down the Control key and typing TAPE-OFF key. Then the TTY may be put ON-LINE again for the next operation.

The paper tape file contents generated by GSFD comes directly from the drum file. It is possible to have character codes on the drum that do not correspond to TTY characters. Those characters will be punched into the tape but will not be printed on the TTY or recognized by the

Assembly Program. The drum files may contain only LF characters between source statements and CR characters at the end of lines. This format may not list well on the TTY which requires LF, CR pairs for inline listing. However, the single LF or single CR is sufficient for the Assembly Program Input and conserve drum space and processing time.

H.6 ABNORMAL HALTS WHEN USING SOURCE FUNCTIONS

Turning the TTY off and on generates noise pulses in the 620i and may result in the processor going into STEP for no apparent reason. The user can restart at S1 or S2 or at START. Usually this will be successful and he may continue at the BRISC. If not, the core image may have to be reloaded from drum or paper tape.

APPENDIX I

TEXT ACCESS COMMANDS

I.1 Text Access Command Word Format

The 620i communicates with the text access controller through a 48 bit interface buffer attached to the I/O bus of the computer. Under program control three 16 bit computer words are transferred to the text access interface to be transmitted to the Text Access Central Station. Words sent to the central station serve as commands to the film and display devices. The elements of the text access commands are as follows:

Device Address: A 6 bit address indicating the console from which the command sequence originated and the function requested. The following codes are current:

BRISC 1: Display 02; Film 01
BRISC 2: Display 06; Film 05

Fiche and Frame Numbers: The retrieval device requires a fiche number in order to select a document from the store and a frame number in order to position a page for either viewing or film reproduction.

Fiche Number: A 12 bit number consisting of 3 BCD digits, each containing four bits. The fiche numbers may range from 000 to 1500 although zero does not correspond to a stored fiche. (The high order digit has a range of 0 to 15 by full use of the four bits and so is not truly a BCD digit.) The fiche number corresponds to the unit record of the retrieval device. That record consists of 60 separate frames. Each frame normally corresponds to a single page of the filmed document.

Frame Number: A 7 bit number consisting of a 3 bit row number concatenated with a 4 bit column number. The fiche contains 60 frames arranged in 5 rows and 12 columns as shown in Fig. I-1. The (1, 1) position would be referenced by the binary string 001 0001; the (5, 12) position by 101 1100.

Number of Frames: A 4 bit number which indicates the number of frames to be generated by the film station when a copy is requested.

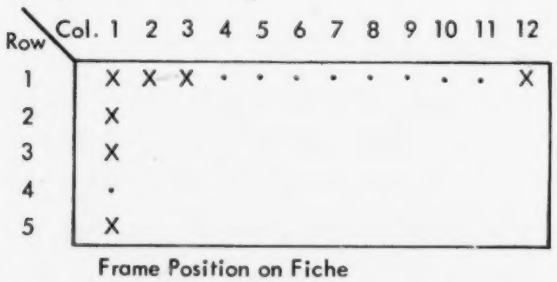


Fig. I.1 Format of Microfiche

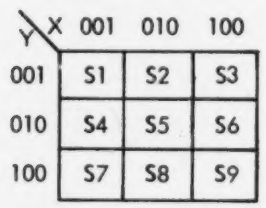


Fig. I.2 Sector Codes

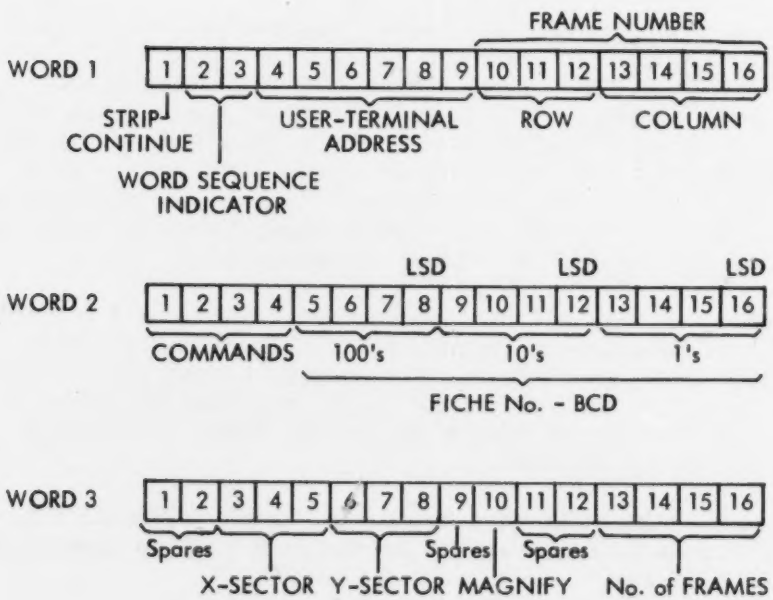


Fig. I.3 Format of Text Command Words

APPENDIX I (Cont'd)

Strip Continue: A 1 bit indicator for the film station. This indicator is not used in the current system and is always set to 1.

Magnify: A 1 bit indicator which is set when requesting that a displayed image be magnified.

X and Y Sector Numbers: Two 3 bit numbers represented in one out of three code used to give the X, Y coordinates of the portion of the page (called the sector) to be magnified. Figure I-2 presents the coordinate representation.

Command Bits: Four bits set aside for future use as special system command codes.

Word Sequence Indicator: A 2 bit code indicating to the text access controller which of four different command word sequences is being issued by the 620i. In the current system a three word command sequence is always transmitted and the Word Sequence Indicator is set to 11.

Figure I-3 shows the format of the three text command words. The first word contains the Text Access Device Address, the Frame Number the Strip Continue bit, and the Word Sequence Indicator. The second word contains the 3 digit BCD fiche number and the 4 bit command code. The third word contains either the 4 bit number of pages count or the S and Y sector numbers with the magnify bit set to one. In the current system if no information is required from the third word the controller ignors its contents; however, the presence of the third word is required in order for the controller to correctly complete its command sequence.

I.2 Display Text Command Sequence

To command the text access system to retrieve a specific page of a stored document and to display it at the storage tube station, the 620i currently formats and outputs a three word sequence to the text controller. The first word contains the terminal address of the display station, the frame number (row and column) of the page to be displayed and a word sequence indicator of 11. The strip continue indicator is not significant in this case and is ignored by the controller. The second word contains the three digit fiche number and the command code is zero. (A zero code is used for all display and copy commands. The device

APPENDIX I (Cont'd)

address determines which command is display and which is copy.) The third word does not contain any information required for this operation; however, its presence is required by the text controller for all operations. The text access controller causes the retrieval device to select the requested fiche and to position the proper frame (page) for display on the storage tube display device.

The retrieval device does not re-store the selected fiche until a new request is made. If the next request is for another frame on the same fiche (as is often the case for a user viewing successive pages of a single document) it is not necessary to re-retrieve the fiche. The text controller hardware checks to determine if a new fiche is required.

APPENDIX J

USE OF DRUM TRACK UTILITY

The Drum Track Utility is used to transfer information between drum tracks and core. Read Track means to take 1024 consecutive words from a given drum track and place them in consecutive core locations at a given address. Write Track transfers 1024 words from core storage to drum storage on a given track.

The Drum Utility requires that the user enter the Drum Sector Command Word into the AR, the Track Word into the BR and the number of tracks into the XR. (For a full discussion of Drum Command Word format see Appendix B. Examples for reading in a full core image is given below.) The UR must be cleared (set to zero) and the Drum Utility entry point placed in the PR.

For Reading into Lower Cor the Intrex Core Image enter the following information:

AR	100000	Sector Word — Read Line 00 into Location 00000.
BR	032000	Track Word — Track Number 52 (decimal)
XR	000004	4 Tracks
UR	000000	Instruction Register = 0
PR	007367	Entry Point for Drum Track Utility

Press System Reset and Run. When the 620i goes into Step enter the following:

AR	100001	Sector Word — Read Line 00 into Location 010000
BR	034000	Track Word — Track Number 56 (decimal)
XR	000004	4 Tracks
UR	000000	Instruction Register = 0
PR	007367	Entry Point for Drum Track Utility

Press System Reset and Run.

The above steps will bring in 8K of 620i words. It is assumed that the given set of tracks is a valid core image which contains a copy of the Drum Track Utility program which overwrites itself.

APPENDIX K

USE OF OCTAL DUMP UTILITY

The Octal Dump Utility is used to output on the Teletype (TTY) a listing of the contents of 620i core locations.

The output listing places the octal location and its contents in octal notation on a single line. The routine requires in the AR the address of the first location to be dumped, in the BR the number of pages to be output and in the XR the number of lines (number of locations) per page.

The following data and steps would result in an octal dump of 256 (decimal) locations beginning at location 02000 (octal) with 64 (0100 octal) lines on each of four pages.

AR	002000	Address of First Location to be dumped.
BR	000004	Number of Pages
XR	000100	Number of Lines Per Page
UR	000000	Instruction Register = 0
PR	007035	Program Starting Address

Turn TTY On-Line

Press System Reset and Run.

After the dump has terminated the 620i halts and A, B and X can be re-loaded for another dump which can be started by pressing Run.

APPENDIX L

START-UP AND LOGIN PROCEDURES

L.1 START-UP 620i AND BRISC

1. Turn the Buffer-Controller on by switching the three power toggle switches on the bottom panel into the up position. The switches should be turned on in sequence from right to left. The corresponding three red power-on lights should light up. Then press the large white power pushbutton light-switch on the 620i front panel.

2. Allow at least a ten-minute warm-up and then open the 620i front panel door and place the Memory Enable/Disable toggle switch in the Enable position. (Switch to the left.)

3. Start the 620i Monitor Program by the following steps:

   a) Use the register display select switch on the 620i front panel to select first the P register (by placing the P switch in the up position) and then press the RESET switch to set the P register contents to zero.

   b) Select and reset the U register.

   c) Press System Reset two or three times.

   d) Press Run.

The above operations begin program control at STRT (Location 00000) which enters the Monitor Initialization program sequence. This action enables the interrupt switches, places the BRISC consoles in Catalog Mode, and enters the Monitor Loop program.

4. The ESL BRISC is turned on by use of the large toggle switch inside the lower right side console door. The warm-up time of the BRISC varies considerably with room temperature. Ten minutes is an average time. (It may be turned on when the Buffer-Controller power is turned on.)

The BRISC may be used to check that the 620i Monitor Program is functioning properly. The following exercises may be performed.

a) Type keyboard letters and observe their appearance on the screen.

b) Change mode from Catalog to Edit and observe the label line change and the Cursor Light come on.

c) Reenter Catalog Mode and press ERAS to erase the page contents.

d) Use the paging buttons to select other display pages. Especially press Next Page, First Page and Previous Page and observe the page number changes. Then press New Page. This tests the function switches, the label read/write routines and the system message read/write routines.

L.2 LOGIN ESL BRISC

Each BRISC has a separate dataphone in the 620i room. The ESL BRISC may be logged in on the BRISC itself by the following:

1. Place the BRISC display on the desired starting page. (Page 1 is a useful starting page.) Press ERAS to clear the page. Place the terminal in Catalog Mode.

2. Dial the BRISC 1 dataphone in the 620i room to 1601. Listen for the dial tone before dialing, then dial. When the ringing stops and the high pitched tone occurs, pull the white plastic hold button on the left side of the handset cradle upwards. Lay the handset down (do not place it back on the cradle).

3. The CTSS greeting should appear on the BRISC screen, terminated with a READY message.

The user types:

login brisc and then the password when it is requested. See the sample dialog in Appendix M.

4. When the login message is complete and CTSS has sent an R (for Ready) message, then the user must call a system program to set the maximum number of lines per page and the maximum number of characters per line. This is done by typing:

r size maxlin 30 Linlth 55

or alternately (if the password is not restricted) by typing,

runcom dis

5. When the next R message appears the user resumes the Intrex retrieval program by first pressing ERAS to get a clean page and then typing:

r intrex hold in.

From here on the system dialog takes over.

L.3 LOGIN BARKER BRISC

The Barker BRISC is logged in remotely from the Teletype (TTY) at the 620i site. After having started the 620i at STRT (Location 00000) in step 3 of the Start-Up 620i Procedure, the Barker BRISC is in Catalog Mode and the interrupt switches are enabled. (This is true even if the BRISC power is not on.) Next the following steps may be taken:

1. Press STEP to halt the 620i.

2. Place the 620i Program Switch 1 on (up position).

3. Place the Teletype ON-LINE.

4. Set the P register to 016000 (octal) and reset the U register to 000000.

5. Press Run (do not press System Reset).

6. Press the TALK button on the BELL SYSTEM dataphone. Listen for a dial tone and then dial 1601. When the ringing stops and a high pitched tone begins, press the DATA button which should light up. Then lay the handset back on the cradle.

7. The CTSS greeting message should be printed on the TTY and the user types on the TTY,

LOGIN BRISC

followed by the password when requested for it.

8. To set the display page size parameters, after receiving a CTSS R (Ready) message, the user types:

R SIZE MAXLIN 30 LINLTH 55

9. After the next R the user resumes the Intrex retrieval programs by typing,

R INTREX HOLD IN

and the Intrex greeting message will appear on the TTY until the page size is exceeded. A delete character ← appears to indicate a CTSS end of page. The user continues the dialog by holding the TTY Control (CTRL) key down and typing an L which issues a FORM (Form Feed) character to CTSS.

The dialog will continue until an Intrex READY appears.

10. The user should issue another FORM character to CTSS to start a new page by holding down the Control (CTRL) key and striking the L key.

11. Next the user presses Step on the 620i to end the Remote Login Sequence and then turns the Teletype OFF.

12. To initialize the Barker BRISC the user sets the P register to 016700 and the U register to 000000, presses System Reset at least twice and then Run twice. The Remote Initialize program sequence begins. It selects user page 1, erases it and then enters the Monitor Initialization sequence and continues on into the Monitor Loop. Both BRISC consoles are now in Catalog Mode and the interrupt switches are enabled for CTSS inputs.

13. Press the remote ATTN/QUIT button on the right side of the TTY keyboard. This will cause a READY message to appear on the Barker BRISC display page 1.

14. The condition of the Barker BRISC may then be observed at the ESL BRISC by placing the Slave Display switch on the ESL BRISC in the Slave position.

L.4 REMOTE LOGOUT OF BARKER BRISC

The logout procedure varies somewhat depending on how the Barker BRISC was left by users. However, some instructions may be helpful.

1. Press Step to halt the 620i.

2. Enter 016000 into the P register and 000000 into the U register. Then press Run. (Do not press System Reset.)

3. Place the TTY On-Line.

4. Check that the 620i Program Switch 1 is in the up position.

5. At this point a variety of actions may be taken at the TTY.

(a) Type a CR and observe the response. If no response comes, type a Form Feed (Control and L Keys). If there is still no response, try the interrupt button (ATTN/QUIT) once. If still no response, try the ATTN/QUIT button twice to generate a system QUIT. If still no response, type the Form Feed again and then repeat the use of the QUIT (press twice). If there is still no response, assume that either CTSS is Down or the terminal has been logged out already. In that case proceed to step 7 below.

If the Intrex program has responded with a READY message type:

QUIT IN

and CTSS should respond with a Thank You message and return control to CTSS command level, finishing with an R (Ready) message.

However, if the system responds with the Please Type Begin message, then the user must use the ATTN/QUIT button. Press it twice until a QUIT message is received followed by an R message. Then type,

OPTION LOAD 0

(the last character before the CR is the numeral zero). This should produce CTSS command level and result in an R message.

6. Having reached the CTSS command level, type; LOGOUT and receive the Time Used message.

7. Press Step on the 620i to halt the program.

8. Lift the Barker BRISC dataphone handset from the cradle and press TALK, then replace the handset.

9. Turn the 620i and Buffer-Controller off as described below.

L.5 BUFFER-CONTROLLER AND 620i TURN OFF

After pressing Step on the 620i to Halt program execution and hanging up data phones;

1. Open the 620i front panel and put the Memory Protect Switch in the DISABLE position.

2. Place the TTY control switch at OFF.

3. Press the 620i Power Switch-Light on the front panel to turn the 620i off.

4. Turn the three Buffer-Controller toggle power switches (located on the bottom panel) to the off (down) position in order from left to right.

5. Turn the ESL BRISC off by use of power toggle switch in the lower right console door. (The BRISC may also be turned off at any time before the Buffer-Controller.)

APPENDIX M

REMOTE LOGIN: CTSS-USER TELETYPE DIALOG

M.1 Login

The following dialog occurred after the 620i Remote Login Sequence had been started and the CTSS dataphone connection dialed in.

```
MIT8CO:  2 USERS AT 05/19/72 1231.3, MAX = 30
READY.
LOGIN BRISC
W 1231.6
PASSWORD
XXXXXX
 STANDBY LINE HAS BEEN ASSIGNED
 YOU HAVE URGENT MAIL
 T0289  9922 LOGGED IN 05/19/72  1231.7 FROM 900008
 LAST LOGOUT WAS 05/18/72  1456.4 FROM 900008
 HOME FILE DIRECTORY IS T0289  CMFL04
****AIS SCHEDULE****
FRIDAY MAY 19, 1972 FROM 1800-2000.
SATURDAY MAY 27, 1972 FROM 1700-1930.
TUESDAY MAY 30, 1972 FROM 1700-1900.
 CTSS BEING USED IS MIT8C0
R 3.066+.000
R SIZE MAXLIN 30 LINLTH 55
W 1232.7
OPX@R .000+.683
R INTREX HOLD IN
W 1232.9
    FOR INFORMATION ABOUT THE RECENT (3/16/72) MODIFICATIONS
TO THE SAVE/USE COMMANDS, TYPE HELP SAVE ←
```

(Output halted at this point until the user typed a Form Feed Character in response to the system end of page indicated by the Delete ← character. Then CTSS output resumed.)

APPENDIX M (Cont'd)

```
AFTER LOGGING IN.

     TO LEARN HOW TO USE THE NEW PAGE COMMAND, TYPE HELP
PAGE AFTER LOGGING IN.
     TO LEARN ABOUT SIZE, TYPE HELP SIZE AFTER LOGGING IN
PLEASE TYPE THE WORD BEGIN FOLLOWED BY A CARRIAGE RETURN.

READY
```

M.2 Logout

The following dialog occurred after a user BRISC session had finished and the 620i Remote Login Sequence restarted. The operator pressed the ATTN/QUIT button to generate a CTSS interrupt response.

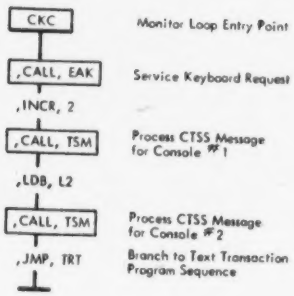
```
INT. 0

READY
QUIT  IN
PASSWORD RECEIVED--       IN
THANK YOU FOR USING INTREX.
R  184.916+69.583
LOGOUT
W  1741.8
  T0289  9922  LOGGED OUT 05/19/72 1741.8 FROM 900008
  TOTAL TIME USED = 4.3 MIN.
```

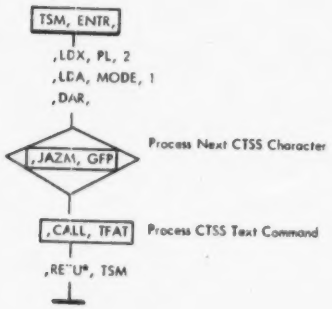
APPENDIX N

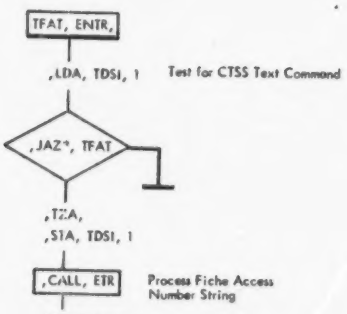
MONITOR PROGRAM FLOW CHARTS

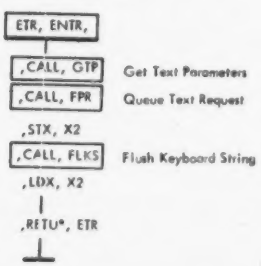
FLOW CHARTS FOR

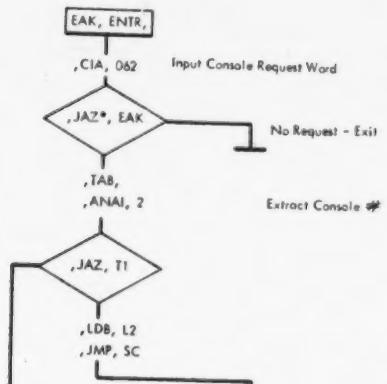
CKC	Check Consoles
EAK	Enter Any Keyboard
TSM	Test Mode
TFAT	Test For Automatic Text
ETR	Enter Text Routine
FLKS	Flush Keyboard String
RPCR	Repeat Core

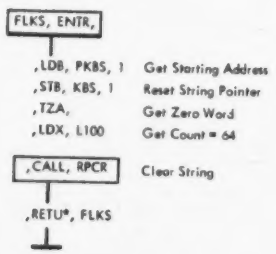

CKC
Monitor Loop Entry Point
,CALL, EAK
Service Keyboard Request
,INCR, 2
,CALL, TSM
Process CTSS Message for Console #1
,LDB, L2
,CALL, TSM
Process CTSS Message for Console #2
,JMP, TRT
Branch to Text Transaction Program Sequence

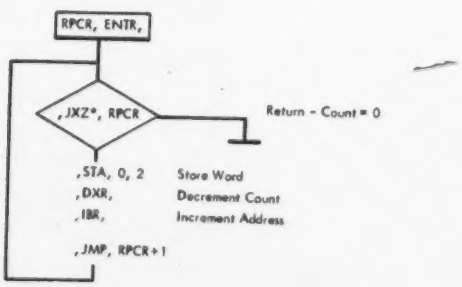

TSM, ENTR,
,LDX, PL, 2
,LDA, MODE, 1
,DAR,
,JAZM, GFP
Process Next CTSS Character
,CALL, TFAT
Process CTSS Text Command
,RETU*, TSM


TFAT, ENTR,
,LDA, TDSI, 1
Test for CTSS Text Command
,JAZ*, TFAT
,TZA,
,STA, TDSI, 1
,CALL, ETR
Process Fiche Access Number String
,RETU*, TFAT


ETR, ENTR,
,CALL, GTP
Get Text Parameters
,CALL, FPR
Queue Text Request
,STX, X2
,CALL, FLKS
Flush Keyboard String
,LDX, X2
,RETU*, ETR


EAK, ENTR,
,CIA, 062
Input Console Request Word
,JAZ*, EAK
No Request - Exit
,TAB,
,ANAI, 2
Extract Console #
,JAZ, T1
,LDB, L2
,JMP, SC
T1, TBA,
, DAR,
,JAZ, C1
,RETU*, EAK
C1, INCR, 2
SC, STB, CONS
Store Console #
, LDX, PL, 2
Get User Parameter Pointer
,CALL, KBI
Get and Process Keyboard Input
,RETU*, EAK

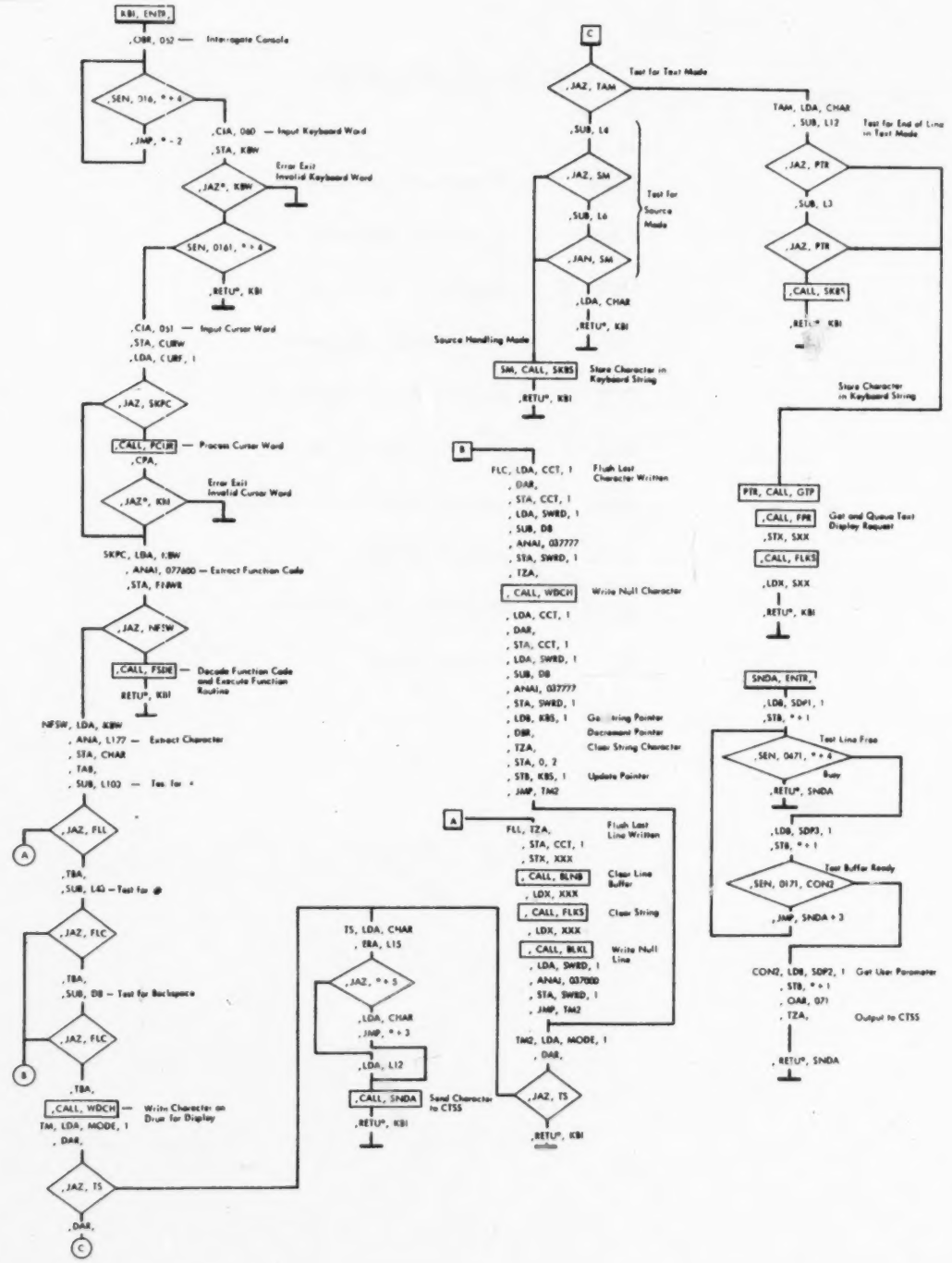

FLKS, ENTR,
,LDB, PKBS, 1
Get Starting Address
,STB, KBS, 1
Reset String Pointer
,TZA,
Get Zero Word
,LDX, L100
Get Count = 64
,CALL, RPCR
Clear String
,RETU*, FLKS


RPCR, ENTR,
,JXZ*, RPCR
Return - Count = 0
,STA, 0, 2
Store Word
,DXR,
Decrement Count
,IBR,
Increment Address
,JMP, RPCR+1

FLOW CHARTS FOR

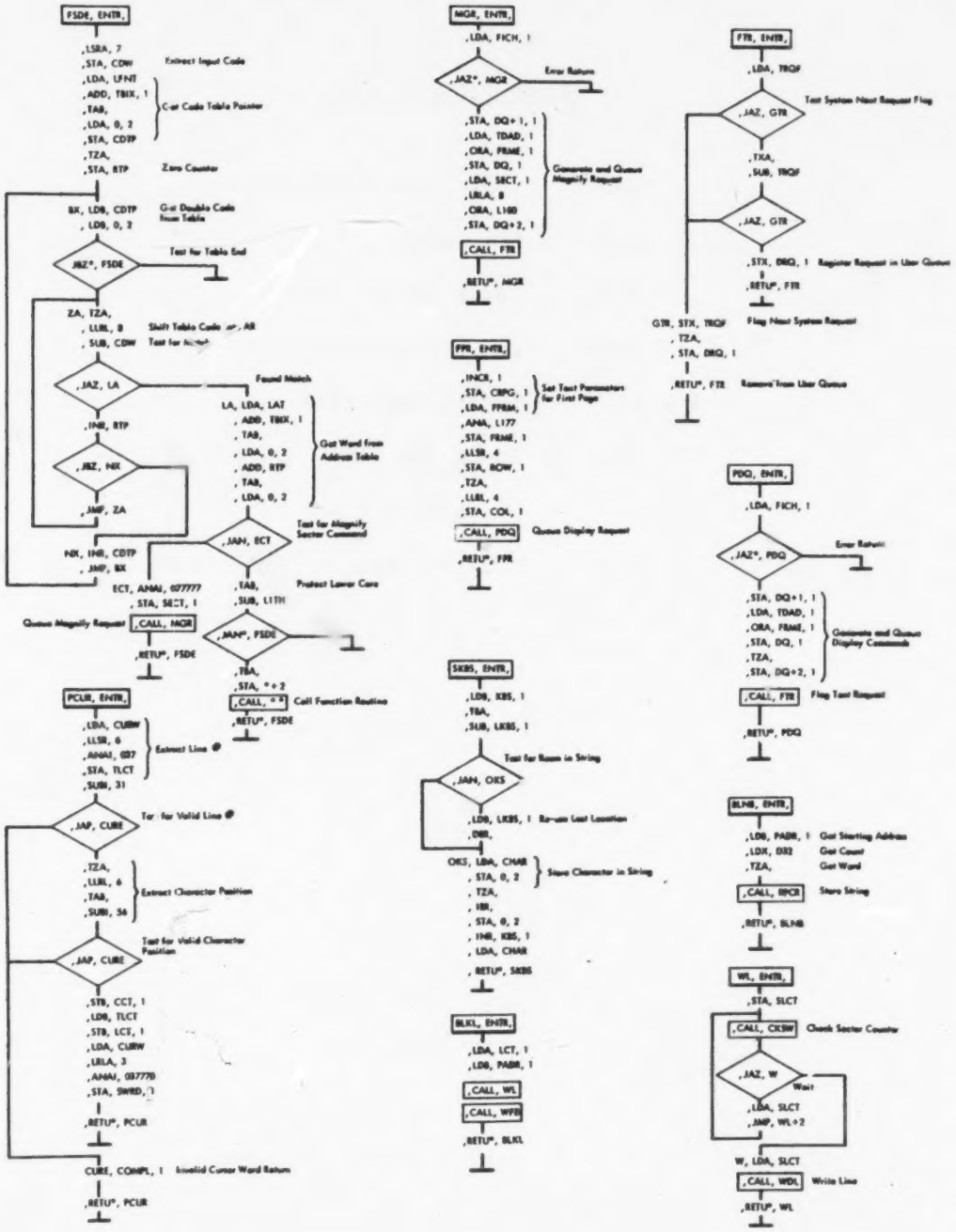
KBI Keyboard Input

SNDA Send A


KBI, ENTR,
,CHR, 052 — Interrogate Console
,SEN, 016, * + 4
,JMP, * - 2
,CIA, 060 — Input Keyboard Word
,STA, KBW
,JAZ*, KBW
Error Exit
Invalid Keyboard Word
,SEN, 0161, * + 4
,RETU*, KBI
,CIA, 051 — Input Cursor Word
,STA, CURW
,LDA, CURF, 1
,JAZ, SKPC
,CALL, PCUR — Process Cursor Word
,CPA,
,JAZ*, KBI
Error Exit
Invalid Cursor Word
SKPC, LDA, KBW
, ANAI, 077600 — Extract Function Code
,STA, FRWR
,JAZ, NFSW
,CALL, FSKE — Decode Function Code and Execute Function Routine
RETU*, KBI
NFSW, LDA, KBW
, ANA, L177 — Extract Character
, STA, CHAR
, TAB,
, SUB, L100 — Test for <
,JAZ, FLL
A
,TBA,
,SUB, L40 — Test for @
,JAZ, FLC
,TBA,
,SUB, D8 — Test for Backspace
,JAZ, FLC
B
,TBA,
,CALL, WDCH — Write Character on Drum for Display
TM, LDA, MODE, 1
, DAR,
,JAZ, TS
,DAR,
C
TS, LDA, CHAR
, ERA, L15
,JAZ, * + 5
,LDA, CHAR
,JMP, * + 3
,LDA, L12
,CALL, SNDA
Send Character to CTSS
,RETU*, KBI
C
,JAZ, TAM
Test for Text Mode
,SUB, L4
,JAZ, SM
,SUB, L6
,JAN, SM
Test for Source Mode
,LDA, CHAR
,RETU*, KBI
Source Handling Mode
SM, CALL, SKBS
Store Character in Keyboard String
,RETU*, KBI
TAM, LDA, CHAR
, SUB, L12
Test for End of Line in Text Mode
,JAZ, PTR
,SUB, L3
,JAZ, PTR
,CALL, SKBS
,RETU* KBI
Store Character in Keyboard String
PTR, CALL, GTP
,CALL, FPR
Get and Queue Text Display Request
,STX, SXX
,CALL, FLKS
,LDX, SXX
,RETU*, KBI
B
FLC, LDA, CCT, 1
Flush Last Character Written
, DAR,
, STA, CCT, 1
, LDA, SWRD, 1
, SUB, D8
, ANAI, 037777
, STA, SWRD, 1
, TZA,
, CALL, WDCH
Write Null Character
, LDA, CCT, 1
, DAR,
, STA, CCT, 1
, LDA, SWRD, 1
, SUB, D8
, ANAI, 037777
, STA, SWRD, 1
, LDB, KBS, 1
Get String Pointer
, DBR,
Decrement Pointer
, TZA,
Clear String Character
, STA, 0, 2
, STB, KBS, 1
Update Pointer
, JMP, TM2
A
FLL, TZA,
Flush Last Line Written
, STA, CCT, 1
, STX, XXX
, CALL, BLNB
Clear Line Buffer
, LDX, XXX
, CALL, FLKS
Clear String
, LDX, XXX
, CALL, BLKL
Write Null Line
, LDA, SWRD, 1
, ANAJ, 037000
, STA, SWRD, 1
, JMP, TM2
TM2, LDA, MODE, 1
, DAR,
,JAZ, TS
,RETU*, KBI
SNDA, ENTR,
,LDB, SDP1, 1
,STB, * + 1
Test Line Free
,SEN, 0471, * + 4
Busy
,RETU*, SNDA
,LDB, SDP3, 1
,STB, * + 1
Test Buffer Ready
,SEN, 0171, CON2
,JMP, SNDA + 3
CON2, LDB, SDP2, 1
Get User Parameter
, STB, * + 1
, OAR, 071
, TZA,
Output to CTSS
, RETU*, SNDA

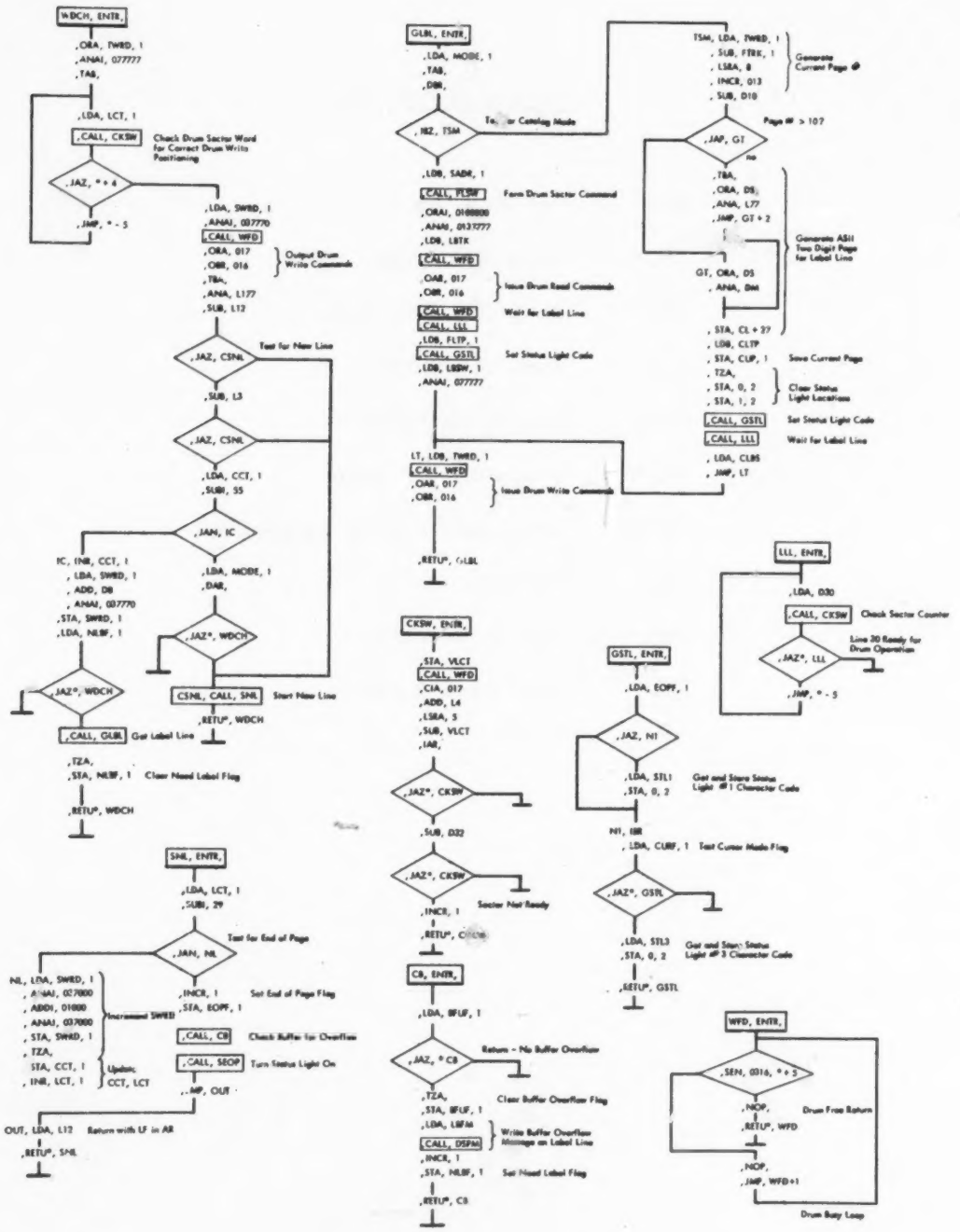
FLOW CHARTS FOR

FSDE	Function Switch Decode
PCUR	Process Cursor
MGR	Magnify Routine
FTR	Flag Text Request
FPR	First Page Routine
PDQ	Put in Drum Queue
SKBS	Store Keyboard String
BLKL	Blank Line
BLNB	Blank Line Buffer
WL	Write Line


FSDE, ENTR,
,LSRA, 7
,STA, CDW
Extract Input Code
,LDA, UFNT
,ADD, TBIX, 1
,TAB,
,LDA, 0, 2
,STA, CDTP
Get Code Table Pointer
,TZA,
,STA, BTP
Zero Counter
BX, LDB, CDTP
, LDB, 0, 2
Get Double Code from Table
,BZE*, FSDE
Test for Table End
ZA, TZA,
, LLBL, 8
, SUB, CDW
Shift Table Code or Test for Match
,JAZ, LA
Found Match
,INR, BTP
,JBZ, NX
,JMP, ZA
LA, LDA, LAT
, ADD, TBIX, 1
, TAB,
, LDA, 0, 2
, ADD, BTP
, TAB,
, LDA, 0, 2
Get Word from Address Table
NX, INR, CDTP
, JMP, BX
,JAN, ECT
Test for Magnify Sector Command
ECT, AMAI, 077777
, STA, SECT, 1
,TAB,
,SUB, L17H
Protect Lower Core
Queue Magnify Request
,CALL, MGR
,RETUP, FSDE
,JMP*, FSDE
,TBA,
,STA, *+2
,CALL, *
Call Function Routine
,RETUP*, FSDE
PCUR, ENTR,
,LDA, CURW
,LLSR, 6
,ANAI, 037
,STA, TLCT
,SUBI, 31
Extract Line #
,JAP, CURE
Test for Valid Line #
,TZA,
,LLBL, 6
,TAB,
,SUBI, 56
Extract Character Position
,JAP, CURE
Test for Valid Character Position
,STB, CCT, 1
,LDB, TLCT
,STB, LCT, 1
,LDA, CURW
,LRLA, 3
,ANAI, 037770
,STA, SWRD, 1
,RETUP, PCUR
CURE, COMPL, 1
Invalid Cursor Word Return
,RETUP*, PCUR
MGR, ENTR,
,LDA, FICH, 1
,JAZ*, MGR
Error Return
,STA, DQ+1, 1
,LDA, TDAD, 1
,ORA, FRME, 1
,STA, DQ, 1
,LDA, SECT, 1
,LRLA, 8
,ORA, L100
,STA, DQ+2, 1
Generate and Queue Magnify Request
,CALL, FTR
,RETUP, MGR
FPR, ENTR,
,INCR, 1
,STA, CRPG, 1
,LDA, FPRM, 1
Set Text Parameters for First Page
,ANA, L177
,STA, FRME, 1
,LLSR, 4
,STA, ROW, 1
,TZA,
,LLBL, 4
,STA, COL, 1
,CALL, PDQ
Queue Display Request
,RETUP, FPR
SKBS, ENTR,
,LDB, KBS, 1
,TBA,
,SUB, LKBS, 1
Test for Room in String
,JAN, OKS
,LDB, LKBS, 1
Re-use Last Location
,DBR,
OKS, LDA, CHAR
, STA, 0, 2
Store Character in String
, TZA,
, IBR,
, STA, 0, 2
, INB, KBS, 1
, LDA, CHAR
, RETUP, SKBS
BLKL, ENTR,
,LDA, LCT, 1
,LDB, PABR, 1
,CALL, WL
,CALL, WFB
,RETUP, BLKL
FTR, ENTR,
,LDA, TRQF
,JAZ, GTR
Test System Next Request Flag
,TXA,
,SUB, TRQF
,JAZ, GTR
,STX, DRQ, 1
Register Request in User Queue
,RETUP, FTR
GTR, STX, TRQF
, TZA,
, STA, DRQ, 1
Flag Next System Request
,RETUP, FTR
Remove from User Queue
PDQ, ENTR,
,LDA, FICH, 1
,JAZ*, PDQ
Error Return
,STA, DQ+1, 1
,LDA, TDAD, 1
,ORA, FRME, 1
,STA, DQ, 1
,TZA,
,STA, DQ+2, 1
Generate and Queue Display Commands
,CALL, FTR
Flag Text Request
,RETUP, PDQ
BLNB, ENTR,
,LDB, PABR, 1
Get Starting Address
,LDX, 032
Get Count
,TZA,
Get Word
,CALL, RPCR
Store String
,RETUP, BLNB
WL, ENTR,
,STA, SLCT
,CALL, CKSW
Check Sector Counter
,JAZ, W
Wait
,LDA, SLCT
,JMP, WL+2
W, LDA, SLCT
,CALL, WDL
Write Line
,RETUP, WL

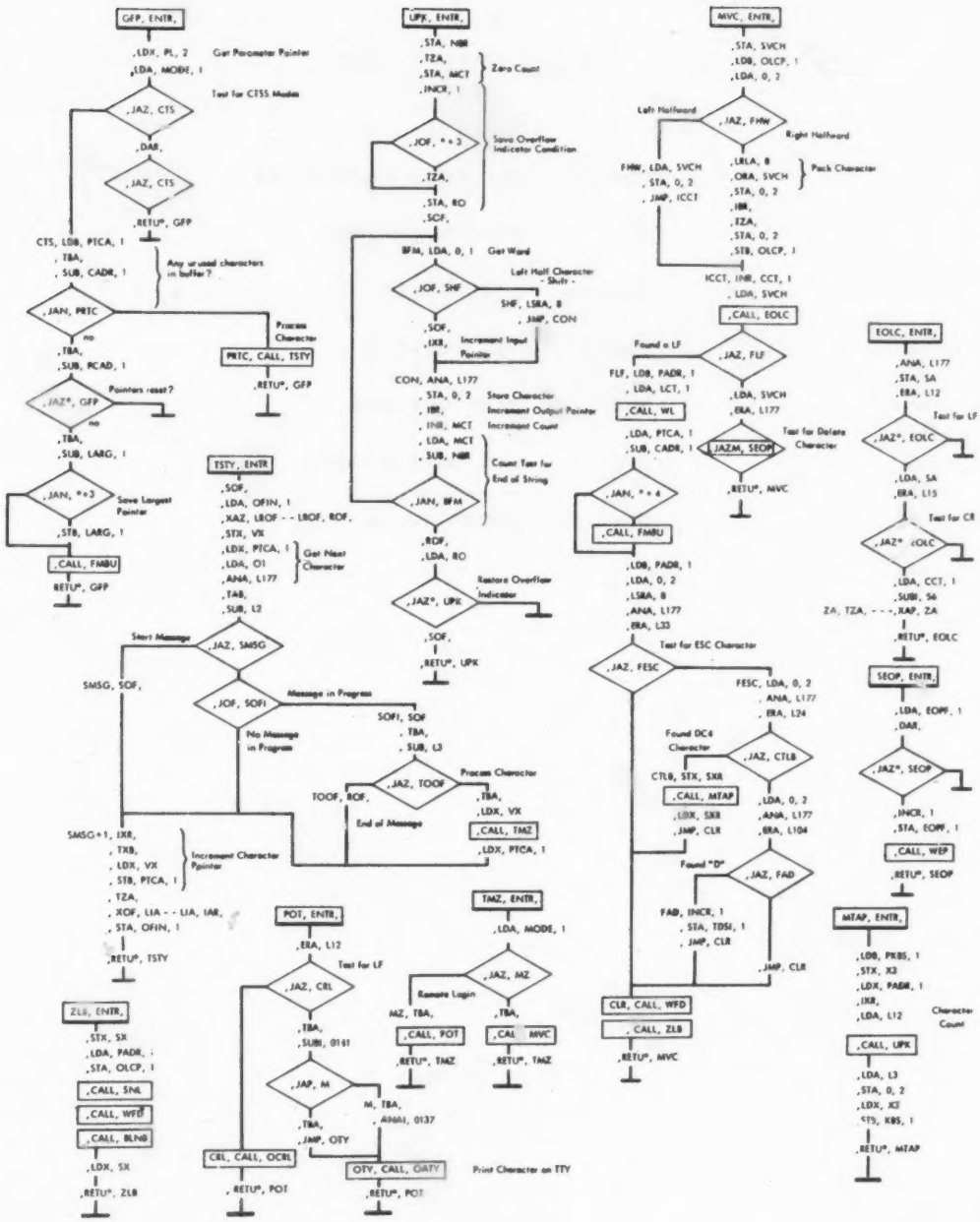
FLOW CHARTS FOR

WDCH	Write Drum Character
GLBL	Get Label
CKSW	Check Sector Word
GSTL	Get Status Lights
LLL	Locate Last Line
SNL	Start New Line
CB	Check Buffer
WFD	Wait For Drum


WDCH, ENTR,
,ORA, TWRD, 1
,ANAI, 077777
,TAB,
,LDA, LCT, 1
,CALL, CKSW
Check Drum Sector Word for Correct Drum Write Positioning
,JAZ, * + 4
,JMP, * - 5
,LDA, SWRD, 1
,ANAI, 037770
,CALL, WFD
,ORA, 017
,OBR, 016
Output Drum Write Commands
,TBA,
,ANA, L177
,SUB, L12
,JAZ, CSNL
Test for New Line
,SUB, L3
,JAZ, CSNL
,LDA, CCT, 1
,SUBI, 55
,JAN, IC
IC, INR, CCT, 1
, LDA, SWRD, 1
, ADD, D8
, ANAI, 037770
,STA, SWRD, 1
,LDA, NLBF, 1
,LDA, MODE, 1
,DAR,
,JAZ*, WDCH
,JAZ*, WDCH
CSNL, CALL, SNL
Start New Line
,RETUP, WDCH
,CALL, GLBL
Get Label Line
,TZA,
,STA, NLBF, 1
Clear Need Label Flag
,RETUP, WDCH
SNL, ENTR,
,LDA, LCT, 1
,SUBI, 29
,JAN, NL
Test for End of Page
NL, LDA, SWRD, 1
, ANAI, 037000
, ADDI, 01000
, ANAI, 037000
, STA, SWRD, 1
, TZA,
STA, CCT, 1
, INR, LCT, 1
Increment SWRD
Update CCT, LCT
,INCR, 1
,STA, EOPF, 1
Set End of Page Flag
,CALL, CB
Check Buffer for Overflow
,CALL, SEOP
Turn Status Light On
,JMP, OUT
OUT, LDA, L12
Return with LF in AR
,RETUP, SNL
GLBL, ENTR,
,LDA, MODE, 1
,TAB,
,DBR,
,NZ, TSM
To... or Catalog Mode
,LDB, SADR, 1
,CALL, FLSW
Form Drum Sector Command
,ORAI, 0100000
,ANAI, 0137777
,LDB, LBTK
,CALL, WFD
,OAB, 017
,OBR, 016
Issue Drum Read Commands
,CALL, WFD
,CALL, LLL
Wait for Label Line
,LDB, FLTP, 1
,CALL, GSTL
Set Status Light Code
,LDB, LBSW, 1
,ANAI, 077777
LT, LDB, TWRD, 1
,CALL, WFD
,OAR, 017
,OBR, 016
Issue Drum Write Commands
,RETUP, GLBL
TSM, LDA, TWRD, 1
, SUB, FTRK, 1
, LSRA, 8
, INCR, 013
, SUB, D10
Generate Current Page #
,JAP, GT
Page # > 107
no
,TBA,
,ORA, D5
,ANA, L77
,JMP, GT + 2
GT, ORA, D5
, ANA, DM
Generate ASII Two Digit Page for Label Line
,STA, CL + 27
,LDB, CLTP
,STA, CLP, 1
Save Current Page
,TZA,
,STA, 0, 2
,STA, 1, 2
Clear Status Light Locations
,CALL, GSTL
Set Status Light Code
,CALL, LLL
Wait for Label Line
, LDA, CL85
, JMP, LT
CKSW, ENTR,
,STA, VLCT
,CALL, WFD
,CIA, 017
,ADD, L4
,LSRA, 5
,SUB, VLCT
,IAR,
,JAZ*, CKSW
,SUB, D32
,JAZ*, CKSW
Sector Not Ready
,INCR, 1
,RETUP, CKSW
CB, ENTR,
,LDA, BFLF, 1
,JAZ, * CB
Return - No Buffer Overflow
,TZA,
,STA, BFLF, 1
Clear Buffer Overflow Flag
,LDA, LBFM
,CALL, DSPM
Write Buffer Overflow Message on Label Line
,INCR, 1
,STA, NLBF, 1
Set Need Label Flag
,RETUP, CB
GSTL, ENTR,
,LDA, EOPF, 1
,JAZ, N1
,LDA, STL1
,STA, 0, 2
Get and Store Status Light #1 Character Code
N1, IBR
, LDA, CURF, 1
Test Cursor Mode Flag
,JAZ*, GSTL
,LDA, STL3
,STA, 0, 2
Get and Store Status Light #3 Character Code
,RETUP, GSTL
LLL, ENTR,
,LDA, D30
,CALL, CKSW
Check Sector Counter
,JAZ*, LLL
Line 30 Ready for Drum Operation
,JMP, * - 5
WFD, ENTR,
,SEN, 0016, * + 3
,NOP,
,RETUP, WFD
Drum Free Return
,NOP,
,JMP, WFD+1
Drum Busy Loop

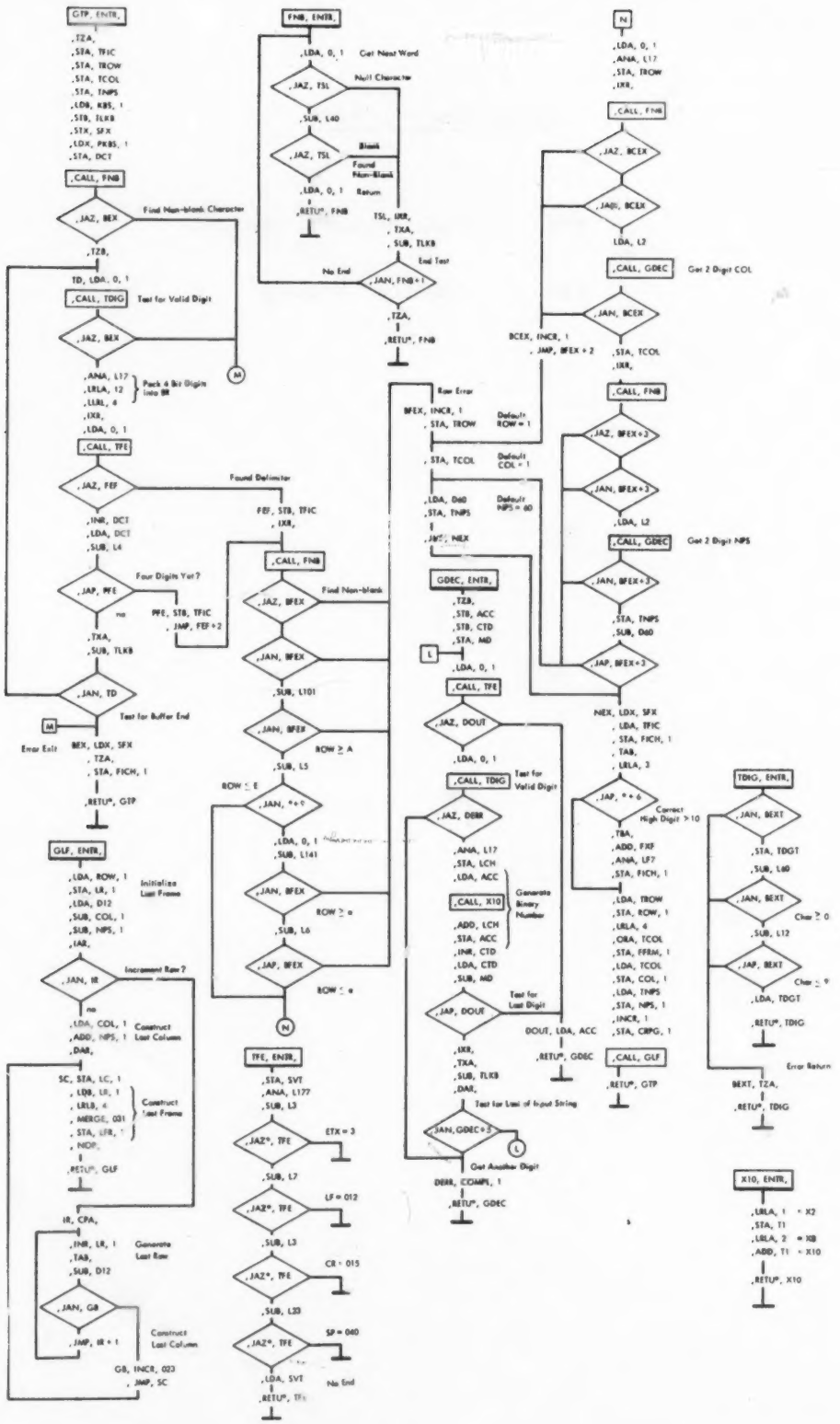
FLOW CHARTS FOR

GFP	Get From Pool
TSTY	Time Share Type
TMZ	Test for Mode Zero
MVC	Move Character
EOLC	End Of Line Check
SEOP	Store End Of Page
MTAP	Move Text Access Parameters
UPK	Unpack
POT	Put On Teletype
ZLB	Zero Line Buffer


GFP, ENTR,
,LDX, PL, 2
Get Parameter Pointer
,LDA, MODE, 1
Test for CTSS Modes
,JAZ, CTS
,DAB,
,JAZ, CTS
,RETUP, GFP
CTS, LDB, PTCA, 1
, TBA,
, SUB, CADR, 1
Any unused characters in buffer?
,JAN, PRTC
no
,TBA,
, SUB, PCAD, 1
Process Character
PRTC, CALL, TSTY
,RETUP, GFP
,JAZP, GFP
Pointers reset?
no
,TBA,
, SUB, LARG, 1
,JAN, *+3
Save Largest Pointer
,STB, LARG, 1
,CALL, FMBU
RETUP, GFP
TSTY, ENTR
,SOF,
,LDA, OFIN, 1
,XAZ, LBOF - - LBOF, ROF,
,STX, VX
,LDX, PTCA, 1
,LDA, 01
,ANA, L177
Get Next Character
,TAB,
,SUB, L2
Start Message
,JAZ, SMSG
SMSG, SOF,
,JOF, SOF1
Message in Progress
No Message in Program
SOF1, SOF
, TBA,
, SUB, L3
,JAZ, TOOF
Process Character
TOOF, ROF,
End of Message
,TBA,
,LDX, VX
,CALL, TMZ
,LDX, PTCA, 1
SMSG+1, IXB,
, TXB,
, LDX, VX
, STB, PTCA, 1
Increment Character Pointer
, TZA,
, XOF, LIA - - LIA, IAR,
, STA, OFIN, 1
,RETUP, TSTY
ZLB, ENTR,
,STX, SX
,LDA, PADR, 1
,STA, OLCP, 1
,CALL, SNL
,CALL, WFD
,CALL, BLNK
,LDX, SX
,RETUP, ZLB
POT, ENTR,
,ERA, L12
Test for LF
,JAZ, CRL
,TBA,
,SUBI, 0101
,JAP, M
M, TBA,
, A96AI, 0137
,TBA,
,JMP, OTY
CRL, CALL, OCRL
, RETUP, POT
OTY, CALL, OATY
Print Character on TTY
,RETUP, POT
TMZ, ENTR,
,LDA, MODE, 1
,JAZ, MZ
Remote Login
MZ, TBA,
,CALL, POT
,RETUP, TMZ
,TBA,
,CAL MVC
,RETUP, TMZ
UPK, ENTR,
,STA, NBR
,TZA,
,STA, MCT
Zero Count
,INCR, 1
,JOF, *+3
Save Overflow Indicator Condition
,TZA,
,STA, RO
,SOF,
BFM, LDA, 0, 1
Get Word
,JOF, SHF
Left Half Character - Shift -
SHF, LSRA, 8
, JMP, CON
,SOF,
,IXB,
Increment Input Pointer
CON, ANA, L177
, STA, 0, 2
Store Character
, IM,
Increment Output Pointer
, INR, MCT
Increment Count
, LDA, MCT
, SUB, NBR
Count Test for End of String
,JAN, BFM
,ROF,
,LDA, RO
,JAZP, UPK
Restore Overflow Indicator
,SOF,
,RETUP, UPK
MVC, ENTR,
,STA, SVCH
,LDB, OLCP, 1
,LDA, 0, 2
,JAZ, FHW
Left Halfword
Right Halfword
FHW, LDA, SVCH
, STA, 0, 2
, JMP, ICCT
,LRLA, 8
,ORA, SVCH
Pack Character
,STA, 0, 2
,IBR,
,TZA,
,STA, 0, 2
,STB, OLCP, 1
ICCT, INR, CCT, 1
, LDA, SVCH
,CALL, EOLC
,JAZ, FLF
Found a LF
FLF, LDB, PADR, 1
, LDA, LCT, 1
,CALL, WL
,LDA, PTCA, 1
,SUB, CADR, 1
,LDA, SVCH
,ERA, L177
,JAZM, SEOP
Test for Delete Character
,RETUP, MVC
,JAN, *+4
,CALL, FMBU
,LDB, PADR, 1
,LDA, 0, 2
,LSRA, 8
,ANA, L177
,ERA, L33
Test for ESC Character
,JAZ, FESC
FESC, LDA, 0, 2
, ANA, L177
, ERA, L24
Found DC4 Character
,JAZ, CTLB
CTLB, STX, SXR
,CALL, MTAP
,LDX, SXR
,JMP, CLR
,LDA, 0, 2
,ANA, L177
,ERA, L104
Found "D"
,JAZ, FAD
FAD, INCR, 1
, STA, TDSI, 1
, JMP, CLR
,JMP, CLR
CLR, CALL, WFD
, CALL, ZLB
,RETUP, MVC
EOLC, ENTR,
,ANA, L177
,STA, SA
,ERA, L12
Test for LF
,JAZP, EOLC
,LDA, SA
,ERA, L15
Test for CR
,JAZ* ,EOLC
,LDA, CCT, 1
,SUBI, 56
ZA, TZA, - - -, XAP, ZA
,RETUP, EOLC
SEOP, ENTR,
,LDA, EOPF, 1
,DAB,
,JAZP, SEOP
,INCR, 1
,STA, EOPF, 1
,CALL, WFP
,RETUP, SEOP
MTAP, ENTR,
,LDB, PKBS, 1
,STX, X3
,LDX, PADR, 1
,IXB,
,LDA, L12
Character Count
,CALL, UPK
,LDA, L3
,STA, 0, 2
,LDX, X3
,STS, KBS, 1
,RETUP, MTAP

FLOW CHARTS FOR

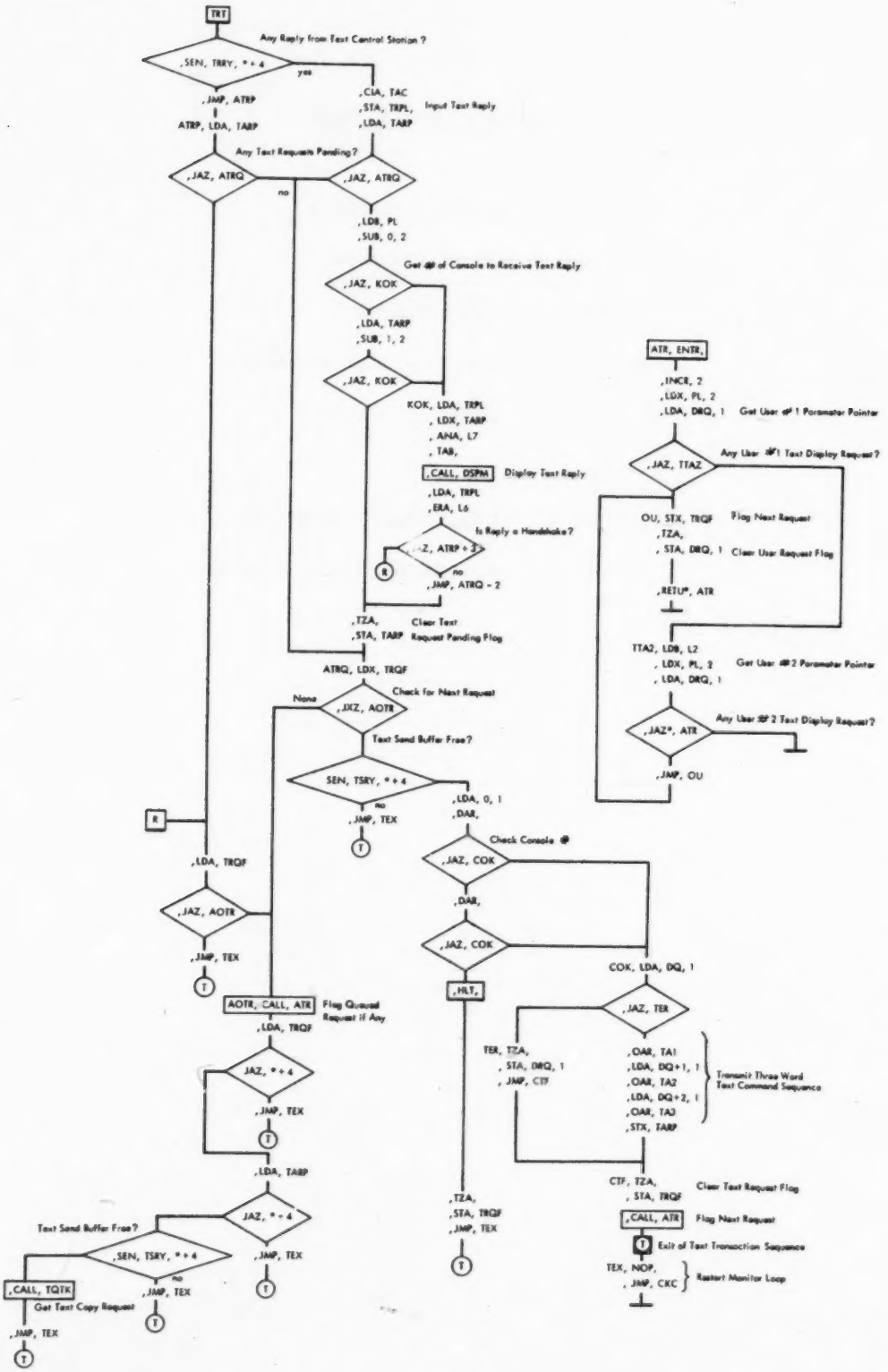
GTP	Get Text Parameters
FNB	Find Non-Blank
TDIG	Test Digit
GDEC	Get Decimal
TFE	Test For End
GLF	Get Last Frame
X10	Times 10


GTP, ENTR,
,TZA,
,STA, TFIC
,STA, TROW
,STA, TCOL
,STA, TNPS
,LDB, KBS, 1
,STB, TLKB
,STX, SFX
,LDX, PKBS, 1
,STA, DCT
,CALL, FNB
,JAZ, BEX
Find Non-blank Character
,TZB,
TD, LDA, 0, 1
,CALL, TDIG
Test for Valid Digit
,JAZ, BEX
,ANA, L17
,LRLA, 12
,LLRL, 4
Pack 4 Bit Digits into BR
,IXB,
,LDA, 0, 1
,CALL, TFE
,JAZ, FEF
Found Delimiter
,INR, DCT
,LDA, DCT
,SUB, L4
,JAP, PFE
Four Digits Yet?
no
PFE, STB, TFIC
,JMP, FEF+2
,TXA,
,SUB, TLKB
,JAN, TD
Test for Buffer End
Error Exit
BEX, LDX, SFX
, TZA,
, STA, FICH, 1
,RETUP, GTP
FEF, STB, TFIC
, IXB,
,CALL, FNB
,JAZ, BFEX
Find Non-blank
,JAN, BFEX
,SUB, L101
,JAN, BFEX
ROW ≥ A
,SUB, L5
,JAN, *+9
ROW ≤ E
,LDA, 0, 1
,SUB, L141
,JAN, BFEX
ROW ≥ 0
,SUB, L6
,JAP, BFEX
ROW < 0
N
TFE, ENTR,
,STA, SVT
,ANA, L177
,SUB, L3
,JAZ*, TFE
ETX = 3
,SUB, L7
,JAZ*, TFE
LF = 012
,SUB, L3
,JAZ*, TFE
CR = 015
,SUB, L23
,JAZ*, TFE
SP = 040
No End
,LDA, SVT
,RETUP, TFE
GLF, ENTR,
,LDA, ROW, 1
,STA, LR, 1
Initialize Last Frame
,LDA, D12
,SUB, COL, 1
,SUB, NPS, 1
,IAR,
,JAN, IR
Increment Row?
no
,LDA, COL, 1
Construct Last Column
,ADD, NPS, 1
,DAR,
SC, STA, LC, 1
, LDB, LR, 1
, LRLB, 4
, MERGE, 031
, STA, LFR, 1
Construct Last Frame
, NOP,
,RETUP, GLF
IR, CPA,
,INR, LR, 1
Generate Last Row
,TAB,
,SUB, D12
,JAN, GB
,JMP, IR + 1
Construct Last Column
GB, INCR, 023
, JMP, SC
FNB, ENTR,
,LDA, 0, 1
Get Next Word
,JAZ, TSL
Null Character
,SUB, L40
,JAZ, TSL
Blank
Found Non-Blank
,LDA, 0, 1
Return
,RETUP, FNB
TSL, INR,
, TXA,
, SUB, TLKB
End Test
No End
,JAN, FNB+1
,TZA,
,RETUP, FNB
N
,LDA, 0, 1
,ANA, L17
,STA, TROW
,IXB,
,CALL, FNB
,JAZ, BCEX
,JAP, BCEX
LDA, L2
,CALL, GDEC
Get 2 Digit COL
,JAN, BCEX
,STA, TCOL
,IXB,
BCEX, INCR, 1
, JMP, BFEX + 2
,CALL, FNB
,JAZ, BFEX+2
,JAN, BFEX+3
,LDA, L2
,CALL, GDEC
Get 2 Digit NPS
,JAN, BFEX+3
,STA, TNPS
,SUB, D60
,JAP, BFEX+3
Row Error
BFEX, INCR, 1
, STA, TROW
Default ROW = 1
, STA, TCOL
Default COL = 1
,LDA, D60
,STA, TNPS
Default NPS = 60
,JMP, NEX
GDEC, ENTR,
,TZB,
,STB, ACC
,STB, CTD
,STA, MD
L
,LDA, 0, 1
,CALL, TFE
,JAZ, DOUT
,LDA, 0, 1
,CALL, TDIG
Test for Valid Digit
,JAZ, DERR
,ANA, L17
,STA, LCH
,LDA, ACC
,CALL, X10
Generate Binary Number
,ADD, LCH
,STA, ACC
,INR, CTD
,LDA, CTD
,SUB, MD
,JAP, DOUT
Test for Last Digit
,IXB,
,TXA,
,SUB, TLKB
,DAR,
Test for Last of Input String
,JAN, GDEC+3
Get Another Digit
DERR, COMPL, 1
,RETUP, GDEC
NEX, LDX, SFX
, LDA, TFIC
, STA, FICH, 1
, TAB,
, LRLA, 3
,JAP, *+6
Correct High Digit >10
,TBA,
,ADD, FXF
,ANA, LF7
,STA, FICH, 1
,LDA, TROW
,STA, ROW, 1
,LRLA, 4
,ORA, TCOL
,STA, FFRM, 1
,LDA, TCOL
,STA, COL, 1
,LDA, TNPS
,STA, NPS, 1
,INCR, 1
,STA, CRPG, 1
DOUT, LDA, ACC
,RETUP, GDEC
,CALL, GLF
,RETUP, GTP
TDIG, ENTR,
,JAN, BEXT
,STA, TDGT
,SUB, L60
,JAN, BEXT
Char ≥ 0
,SUB, L12
,JAP, BEXT
Char ≤ 9
,LDA, TDGT
,RETUP, TDIG
Error Return
BEXT, TZA,
,RETUP, TDIG
X10, ENTR,
,LRLA, 1 = X2
,STA, T1
,LRLA, 2 = X8
,ADD, T1 = X10
,RETUP, X10

FLOW CHARTS FOR

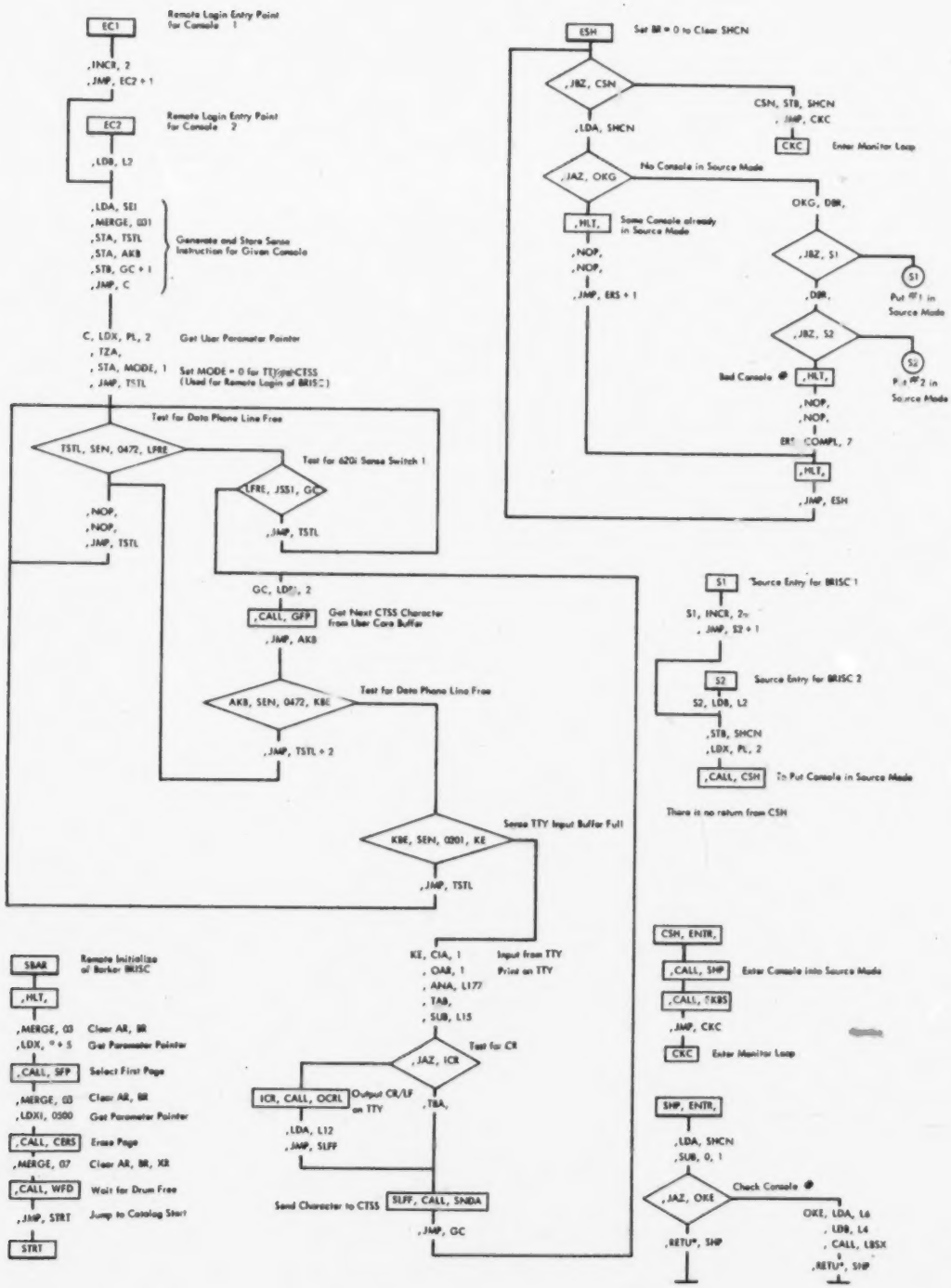
TRT Text Request Test

ATR Any Text Request


TRT
Any Reply from Text Control Station ?
,SEN, TRRY, * + 4
yes
,JMP, ATRP
ATRP, LDA, TARP
,CIA, TAC
,STA, TRPL,
,LDA, TARP
Input Text Reply
Any Text Requests Pending?
,JAZ, ATRQ
,JAZ, ATBQ
no
,LDB, PL
,SUB, 0, 2
Get # of Console to Receive Text Reply
,JAZ, KOK
,LDA, TARP
,SUB, 1, 2
,JAZ, KOK
KOK, LDA, TRPL
, LDX, TARP
, ANA, L7
, TAB,
,CALL, DSPM
Display Text Reply
,LDA, TRPL
,ERA, L6
Is Reply a Handshake?
,Z, ATRP + 2
R
no
,JMP, ATRQ - 2
,TZA,
,STA, TARP
Clear Text Request Pending Flag
ATRQ, LDX, TRQF
None
Check for Next Request
,JXZ, AOTR
Text Send Buffer Free?
SEN, TSRY, * + 4
no
,JMP, TEX
T
R
,LDA, TRQF
,JAZ, AOTR
,JMP, TEX
T
AOTR, CALL, ATR
Flag Queued Request If Any
,LDA, TRQF
JAZ, * + 4
,JMP, TEX
T
,LDA, TARP
JAZ, * + 4
Text Send Buffer Free?
,SEN, TSRY, * + 4
no
,JMP, TEX
,JMP, TEX
T
,CALL, TQTK
Get Text Copy Request
,JMP, TEX
T
T
,LDA, 0, 1
,DAR,
Check Console #
,JAZ, COK
,DAR,
,JAZ, COK
,HLT,
,TZA,
,STA, TRQF
,JMP, TEX
T
COK, LDA, DQ, 1
,JAZ, TER
TER, TZA,
, STA, DRQ, 1
, JMP, CTF
,OAR, TA1
,LDA, DQ+1, 1
,OAR, TA2
,LDA, DQ+2, 1
,OAR, TA3
,STX, TARP
Transmit Three Word Text Command Sequence
CTF, TZA,
, STA, TRQF
Clear Text Request Flag
,CALL, ATR
Flag Next Request
T
Exit of Text Transaction Sequence
TEX, NOP,
, JMP, CKC
Restart Monitor Loop
ATR, ENTR,
,INCR, 2
,LDX, PL, 2
,LDA, DRQ, 1
Get User #1 Parameter Pointer
,JAZ, TTAZ
Any User #1 Text Display Request?
OU, STX, TRQF
Flag Next Request
,TZA,
, STA, DRQ, 1
Clear User Request Flag
,RETU*, ATR
TTA2, LDB, L2
, LDX, PL, 2
, LDA, DRQ, 1
Get User #2 Parameter Pointer
,JAZ*, ATR
Any User #2 Text Display Request?
,JMP, OU

FLOW CHARTS FOR

EC1	Enter Console 1
EC2	Enter Console 2
SBAR	Start Barker
ESH	Enter Source Handling
S1	Source 1
S2	Source 2
CSH	Call Source Handling
SHP	Source Handling Program

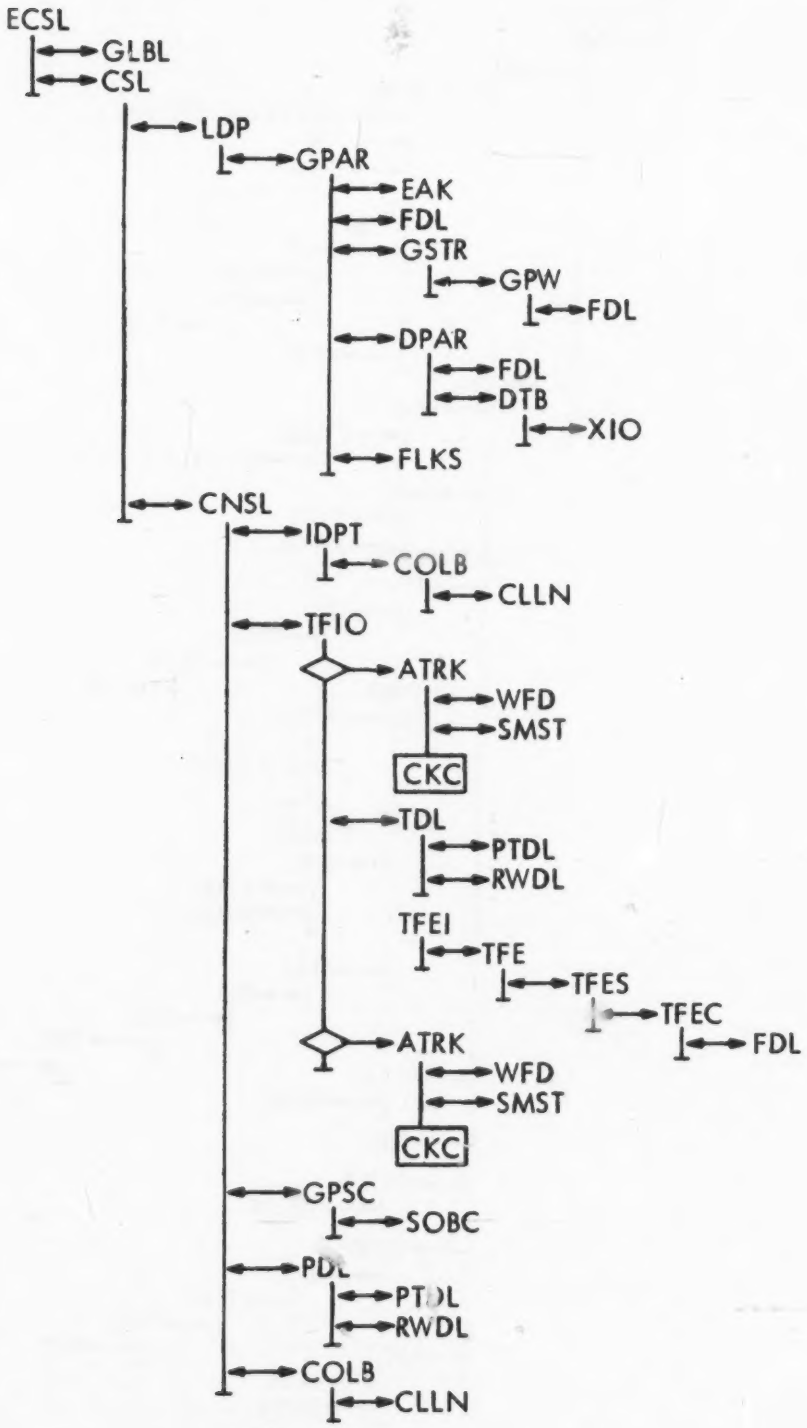

EC1
Remote Login Entry Point for Console 1
,INCR, 2
,JMP, EC2 + 1
EC2
Remote Login Entry Point for Console 2
,LDB, L2
,LDA, SEI
,MERGE, 001
,STA, TSTL
,STA, AKB
,STB, GC + 1
,JMP, C
Generate and Store Sense Instruction for Given Console
C, LDX, PL, 2
, TZA,
, STA, MODE, 1
, JMP, TSTL
Get User Parameter Pointer
Set MODE = 0 for TTY mode CTSS (Used for Remote Login of BRISC)
Test for Data Phone Line Free
TSTL, SEN, 0472, LFRE
Test for 620i Sense Switch 1
LFRE, JSS1, GC
,NOP,
,NOP,
,JMP, TSTL
,JMP, TSTL
GC, LDB, 2
,CALL, GFP
Get Next CTSS Character from User Core Buffer
,JMP, AKB
AKB, SEN, 0472, KBE
Test for Data Phone Line Free
,JMP, TSTL + 2
KBE, SEN, 0001, KE
Sense TTY Input Buffer Full
,JMP, TSTL
ESH
Set BR = 0 to Clear SHCN
,JBZ, CSN
CSN, STB, SHCN
, JMP, CKC
,LDA, SHCN
CKC
Enter Monitor Loop
,JAZ, OKG
No Console in Source Mode
OKG, DBR,
,HLT,
Some Console already in Source Mode
,JBZ, S1
,NOP,
,NOP,
,JMP, ERS + 1
,DBR,
S1
Put #1 in Source Mode
,JBZ, S2
Bad Console #
,HLT,
S2
Put #2 in Source Mode
,NOP,
,NOP,
ERR, COMPL, 7
,HLT,
,JMP, ESH
S1
Source Entry for BRISC 1
S1, INCR, 2
, JMP, S2 + 1
S2
Source Entry for BRISC 2
S2, LDB, L2
,STB, SHCN
,LDX, PL, 2
,CALL, CSH
To Put Console in Source Mode
There is no return from CSH
SBAR
Remote Initialize of Barker BRISC
,HLT,
,MERGE, 03
Clear AR, BR
,LDX, * + 5
Get Parameter Pointer
,CALL, SFP
Select First Page
,MERGE, 03
Clear AR, BR
,LDXI, 0500
Get Parameter Pointer
,CALL, CBRS
Erase Page
,MERGE, 07
Clear AR, BR, XR
,CALL, WFD
Wait for Drum Free
,JMP, STRT
Jump to Catalog Start
STRT
KE, CIA, 1
Input from TTY
, OAR, 1
Print on TTY
, ANA, L177
, TAB,
, SUB, L15
,JAZ, ICR
Test for CR
ICR, CALL, OCRL
Output CR/LF on TTY
,TBA,
,LDA, L12
,JMP, SLFF
Send Character to CTSS
SLFF, CALL, SNDA
,JMP, GC
CSH, ENTR,
,CALL, SHP
Enter Console into Source Mode
,CALL, TKBS
,JMP, CKC
CKC
Enter Monitor Loop
SHP, ENTR,
,LDA, SHCN
,SUB, 0, 1
,JAZ, OKE
Check Console #
,RETU*, SHP
OKE, LDA, L6
, LDB, L4
, CALL, LBSX
,RETU*, SHP

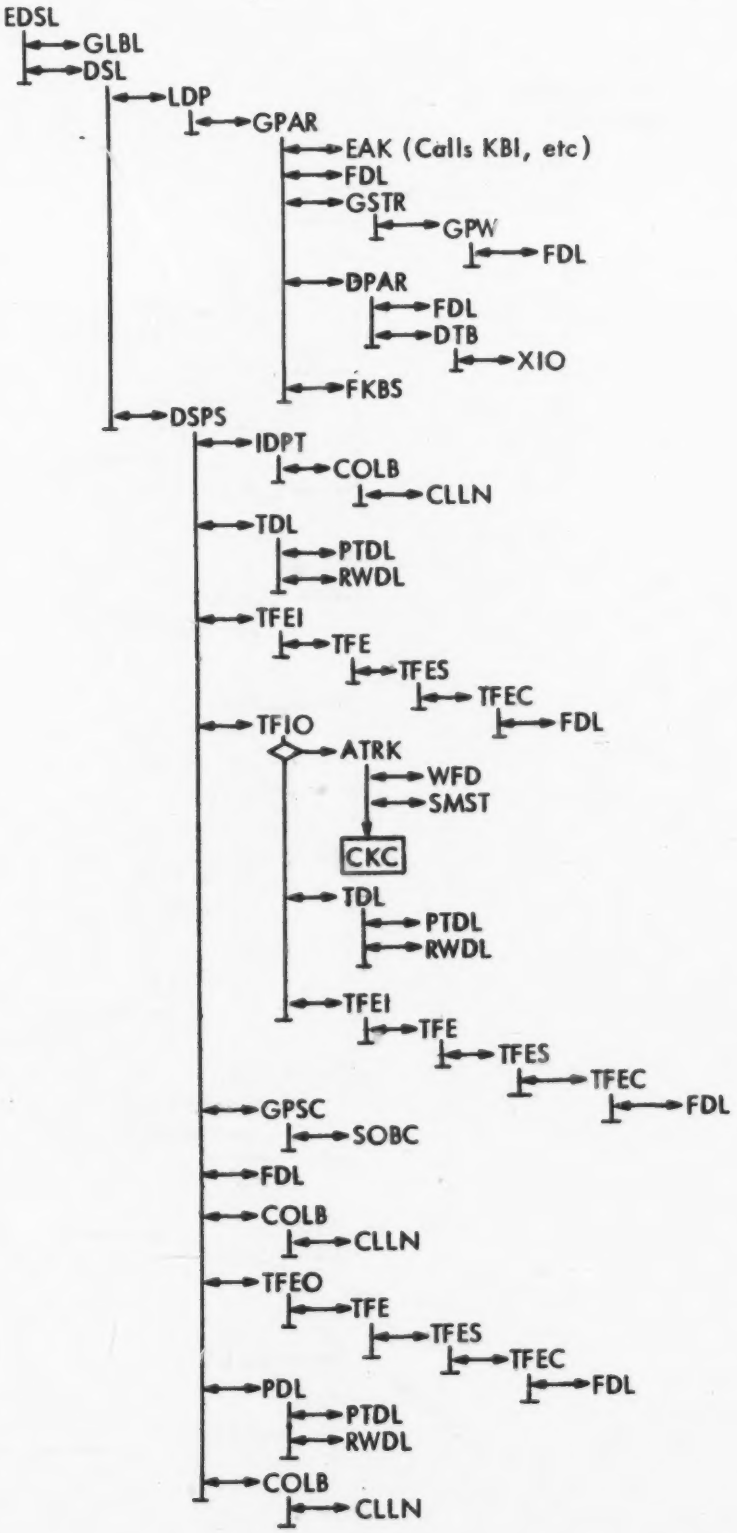
APPENDIX O

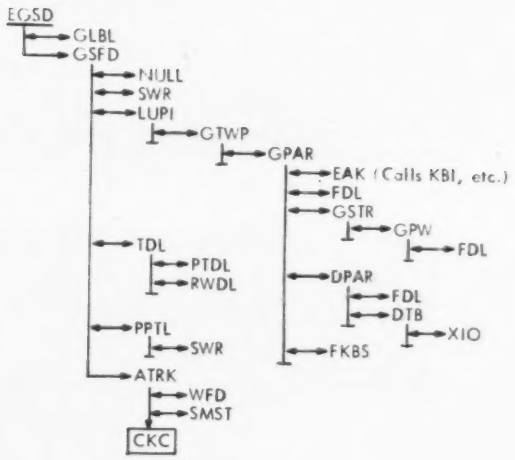
SOURCE HANDLING CALLING SEQUENCES AND SELECTED FLOW CHARTS

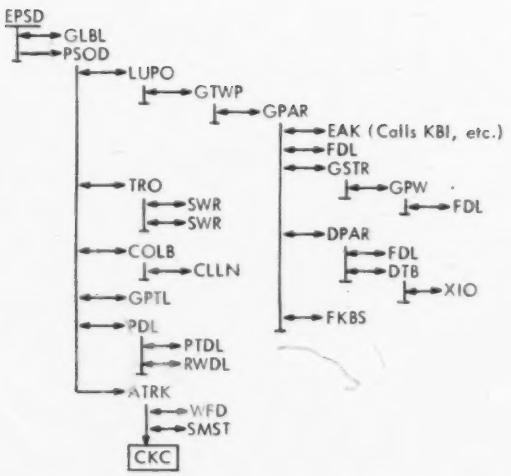
CALLING SEQUENCES FOR

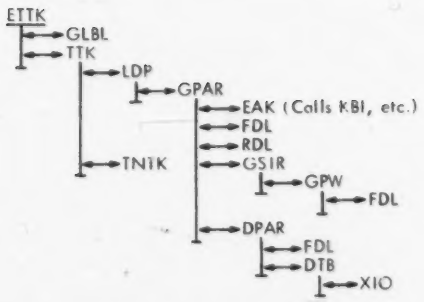
ECSL	Enter Condense Source Lines
EDSL	Enter Display Source Lines
EGSD	Enter Get Source From Drum
EPSD	Enter Put Source On Drum
ETTK	Enter Transfer Tracks


ECSL
GLBL
CSL
LDP
GPAR
EAK
FDL
GSTR
GPW
FDL
DPAR
FDL
DTB
XIO
FLKS
CNSL
IDPT
COLB
CLLN
TFIO
ATRK
WFD
SMST
CKC
TDL
PTDL
RWDL
TFEI
TFE
TFES
TFEC
FDL
ATRK
WFD
SMST
CKC
GPSC
SOBC
PDL
PTDL
RWDL
COLB
CLLN


EDSL
GLBL
DSL
LDP
GPAR
EAK (Calls KBI, etc)
FDL
GSTR
GPW
FDL
DPAR
FDL
DTB
XIO
FKBS
DSPS
IDPT
COLB
CLLN
TDL
PTDL
RWDL
TFEI
TFE
TFES
TFEC
FDL
TFIO
ATRK
WFD
SMST
CKC
TDL
PTDL
RWDL
TFEI
TFE
TFES
TFEC
FDL
GPSC
SOBC
FDL
COLB
CLLN
TFEO
TFE
TFES
TFEC
FDL
PDL
PTDL
RWDL
COLB
CLLN

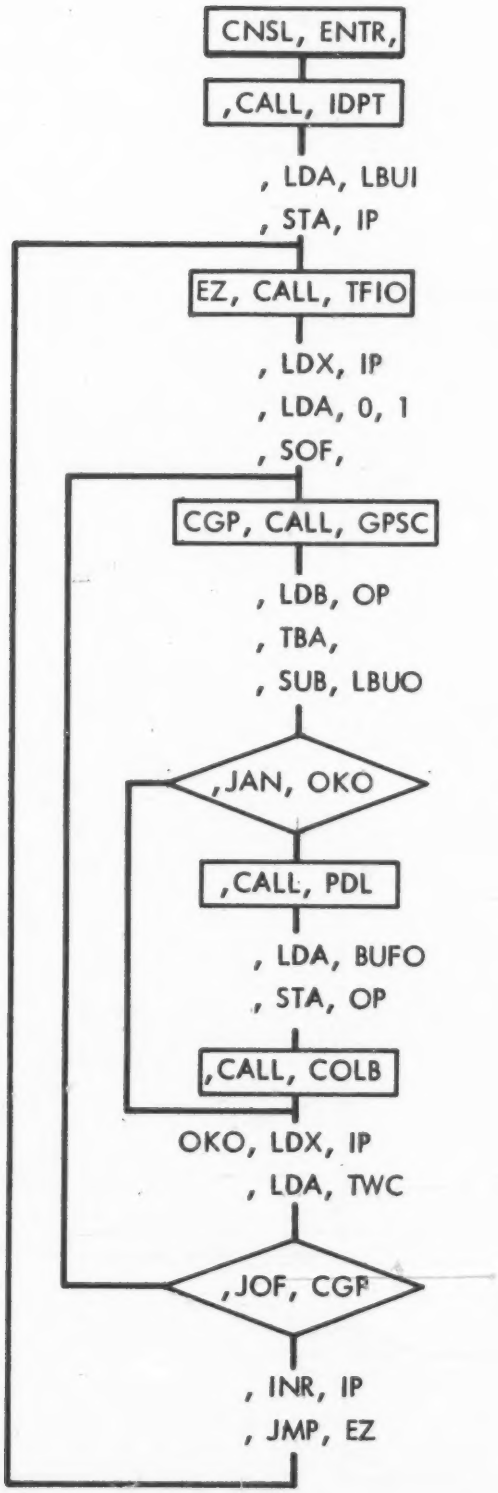

EGSD
GLBL
GSFD
NULL
SWR
LUPI
GTWP
GPAR
EAK (Calls KBI, etc.)
FDL
GSTR
GPW
FDL
TDL
PTDL
RWDL
DPAR
FDL
DTB
XIO
PPTL
SWR
FKBS
ATRK
WFD
SMST
CKC

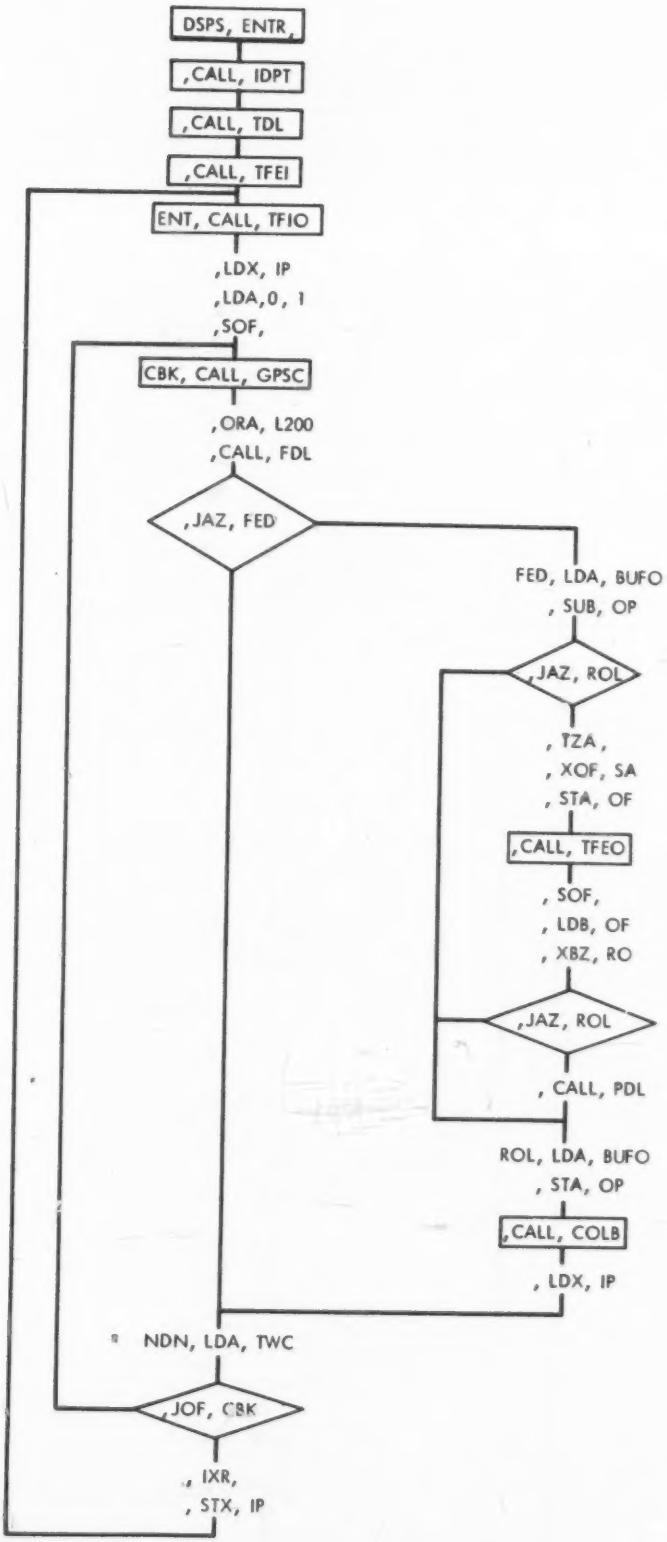

EPSD
GLBL
PSOD
LUPO
GTWP
GPAR
EAK (Calls KBI, etc.)
FDL
GSTR
GPW
FDL
TRO
SWR
SWR
DPAR
FDL
DTB
XIO
COLB
CLLN
FKBS
GPTL
PDL
PTDL
RWDL
ATRK
WFD
SMST
CKC

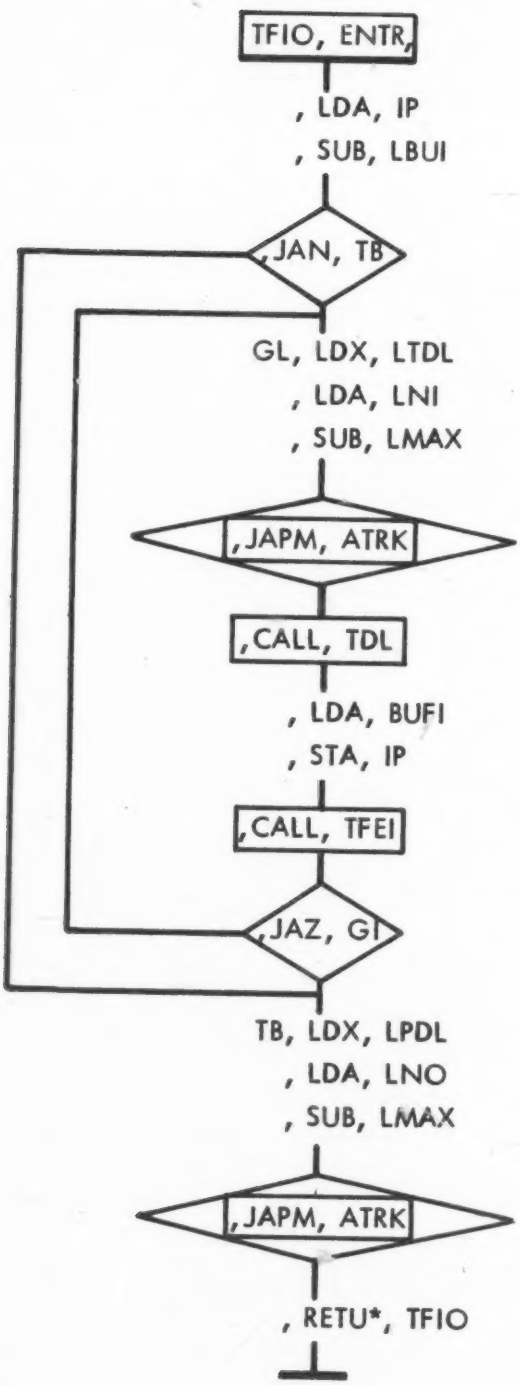

ETTK
GLBL
TTK
LDP
GPAR
EAK (Calls KBI, etc.)
FDL
RDL
TNTK
GSTR
GPW
FDL
DPAR
FDL
DTB
XIO

FLOW CHARTS FOR

CNSL	Condense Source Lines
DSPS	Display Source
TFIO	Test For Input/Output


CNSL, ENTR,
,CALL, IDPT
, LDA, LBUI
, STA, IP
EZ, CALL, TFIO
, LDX, IP
, LDA, 0, 1
, SOF,
CGP, CALL, GPSC
, LDB, OP
, TBA,
, SUB, LBUO
,JAN, OKO
,CALL, PDL
, LDA, BUFO
, STA, OP
,CALL, COLB
OKO, LDX, IP
, LDA, TWC
,JOF, CGP
, INR, IP
, JMP, EZ


DSPS, ENTR,
,CALL, IDPT
,CALL, TDL
,CALL, TFEI
ENT, CALL, TFIO
,LDX, IP
,LDA,0, 1
,SOF,
CBK, CALL, GPSC
,ORA, L200
,CALL, FDL
,JAZ, FED
FED, LDA, BUFO
, SUB, OP
,JAZ, ROL
, TZA,
, XOF, SA
, STA, OF
,CALL, TFEO
, SOF,
, LDB, OF
, XBZ, RO
,JAZ, ROL
, CALL, PDL
ROL, LDA, BUFO
, STA, OP
,CALL, COLB
, LDX, IP
NDN, LDA, TWC
,JOF, CBK
, IXR,
, STX, IP


TFIO, ENTR,
, LDA, IP
, SUB, LBUI
,JAN, TB
GL, LDX, LTDL
, LDA, LNI
, SUB, LMAX
,JAPM, ATRK
,CALL, TDL
, LDA, BUFI
, STA, IP
,CALL, TFEI
,JAZ, GI
TB, LDX, LPDL
, LDA, LNO
, SUB, LMAX
,JAPM, ATRK
, RETU*, TFIO

APPENDIX P

CROSS REFERENCES OF ENTRY POINTS AND SUBROUTINES

P.1 Entry Points

Entry Point	Location	Paragraph
CKC	01765	4.3.3.1
CRWI	07367	4.3.6.1.1
DNPG	07035	4.3.6.2.1
ECAT	01012	4.3.2.1
ELOC	01000	4.3.2.2
S1	12027	N.8
S2	12032	N.8
SBAR	16700	4.3.4.3
TEX	11311	N.1
TRT	11237	4.3.3.2

P.2 General Subroutines

Routine	Location	Paragraph
ATR	11354	4.4.9.1
BLKL	02466	4.4.1.1
BLNB	02102	4.4.3.1
BSLB	10120	4.4.3.2
CATM	01033	4.4.7.1
CB	10630	4.4.1.2
CERS	01665	4.4.1.3
CFVE	16432	4.4.7.2
CKSW	01416	4.4.2.1
CR	11500	4.4.9.2
CSVP	10334	4.4.8.1
CTC	11701	4.4.4.1
CURM	01042	4.4.2.3
DMPO	07041	4.4.11.1
DSPM	10000	4.4.6.1
EAK	01077	4.4.1.4
EDIT	10300	4.4.7.4
EL	10307	4.4.1.5
EMG	11232	4.4.9.3
EOLC	02252	4.4.1.6
ERAS	01701	4.4.1.7
EROP	10500	4.4.8.2
ETA	03100	4.4.9.4
ETR	03243	4.4.9.5

Calls	Called By
—	—
WL	KBI
RPCR	ZLB,KBI
RPCR	EL,ERAS
SMST	ECAT
DSPM	SNL
ERAS,SNDA,DSPM	(FSDE),GNWP
RSTP,NCAT	(FSDE)
WFD	WSC,WDCH,LLL,WL,GM,WM,WFL
TQPU	(FSDE)
—	(FSDE)
—	QPOP
GLBL	(FSDE)
OCRL,SPA5,ODMP,RPTO	LD
GM,WFD,GSTL,WM	CB,GNWP,CERS,IDFR,TQPU
KBI	GPAR
SMST	(FSDE),NLOC
BSLB,WL	(FSDE)
—	(FSDE)
—	MVC
BSLB,WXL,GLBL,WFD,ZSWR,FLKS	EROP,CERS
ERAS	(FSDE)
SMST	(FSDE)
GTP,FPR,FLKS	TFAT

Routine	Location	Paragraph
ETS1	00600	4.4.5.8
ETS2	00605	4.4.5.8
FF5L	07151	4.4.11.2
FLKS	02455	4.4.3.3
FLSW	01515	4.4.2.2
FMBU	02072	4.4.5.1
FNB	03612	4.4.9.6
FPFV	16443	4.4.7.5
FPR	11166	4.4.9.7
FSDE	01536	4.4.1.8
FTR	11152	4.4.9.8
GDEC	03655	4.4.9.9
GFP	01777	4.4.5.2
GLBL	01440	4.4.1.9
GLF	11731	4.4.9.10
GM	10013	4.4.6.2
GNWP	16300	4.4.1.10
GSTL	02660	4.4.1.11
GTP	03400	4.4.9.11
IDFR	11025	4.4.9.12
IZTQ	11567	4.4.9.13
KBI	01150	4.4.1.12
LD	07012	4.4.11.3
LLL	01503	4.4.2.3
LOCM	01024	4.4.7.6

Calls	Called By
—	DPI1
—	DPI2
RPTO	LD
RPCR	ERAS,KBI,ETR
—	GLBL,WDL,GM,WM,SVL,RTL
—	GFP
—	GTP
RSTP,SFP	(FSDE)
PDQ	KBI,ETR
MGR	KBI
—	PDQ,MGR
TFE,TDIG,X10	GTP
FMBU,TSTY	TSM
FLSW,WFD,LLL,GSTL	TSLB,WDCH,ERAS,CURM
—	GTP
CKSW,FLSW,WFD	DSPM
SNP,DSPM,CERS	(FSDE)
—	GLBL,DSPM
FNB,TDIG,TFE,GDEC,GLF	KBI,ETR
DSPM	QDR
—	TQTK
PCUR,FSDE,WDCH,SKBS,SNDA,BLNB, FLKS,BLKL,GTP,FPR	EAK
DMPO,FF5L	—
CKSW	GLBL
SMST	—

Routine	Location	Paragraph
LPR	11410	4.4.9.14
MGR	11212	4.4.9.15
MTAP	03257	4.4.9.16
MVC	02170	4.4.5.3
NCAT	16420	4.4.7.7
NLOC	16400	4.4.7.8
NPR	11000	4.4.9.17
NTLN	16530	4.4.1.13
NULL	06125	4.4.11.4
OATY	07227	4.4.11.5
OCRL	07237	4.4.11.6
ODMP	07200	4.4.11.7
OTCH	07245	4.4.11.8
PCUR	16630	4.4.1.14
PDQ	11134	4.4.9.18
POT	16060	4.4.5.4
PPR	11006	4.4.9.19
QDR	11014	4.4.9.20
QPOP	11550	4.4.9.21
RFY	16454	4.4.7.9
RPCR	01140	4.4.4.2
RPTO	07270	4.4.11.9
RSTP	16344	4.4.7.10
RTL	10404	4.4.8.3
RWDL	07325	4.4.10.1
RWDT	07353	4.4.10.2

Calls	Called By
PDQ	(FSDE)
FTR	FSDE
UPK	MVC
EOLC,SEOP,WEP,WL,WFD,ZLB,MTAP	TMZ
SMST	(FSDE),CFVE
VPAR,EDIT	(FSDE)
QDR	(FSDE)
SVL,VPAR,PCUR,RSTP	(FSDE)
SWR	GSFD
—	POT,ODMP,RPTO
OTCH	POT,DMPO
OATY	DMPO
—	OCRL
—	KBI,NTLN
FTR	FPR,SPR,LPR,QDR
OCRL,OATY	TMZ
QDR	(FSDE)
IDFR,PDQ	NPR,PPR
CTC	TQPU
RSTP,SMST	(FSDE)
—	BLNB,FLKS,BSLB,FKBS
OATY	FF5L,SPA5,DMPO
—	CFVE,RFV,NTLN,RSTP
TFML,WFL,FLSW,WDL	(FSDE)
—	RWDT,PDL,TDL
RWDL	RWI

Routine	Location	Paragraph
RWI	07341	4.4.10.3
SEOP	10765	4.4.1.15
SFP	01605	4.4.1.16
SKBS	02477	4.4.1.17
SMST	01057	4.4.1.18
SNDA	02055	4.4.5.5
SNL	01364	4.4.1.19
SNP	01614	4.4.1.20
SPA5	07216	4.4.11.10
SPP	01641	4.4.1.21
SPR	11205	4.4.9.22
SVL	10342	4.4.8.4
SWR	06154	4.4.11.11
TDIG	03634	4.4.9.23
TFAT	03231	4.4.9.24
TFE	03567	4.4.9.25
TFML	10323	4.4.1.22
TMZ	16100	4.4.5.6
TQPU	11505	4.4.9.26
TQTK	11600	4.4.9.27
TSLB	01067	4.4.1.23
TSM	03140	4.4.1.24
TSTY	02112	4.4.5.7
UPK	02540	4.4.4.3
VMDE	16466	4.4.7.11
VPAR	16320	4.4.7.12

Calls	Called By
RWDT	TNTK
WEP	MVC,SNL
TSLB	(FSDE),SNP,FPFV
—	KBI
TSLB	EDIT,ETA,NCAT,RFV,VMDE,VSMP, CATM,LOCM,LBSX,ATRK,EATK
—	KBI,CERS
CB,SEOP	WDCH,ZLB
SFP,TSLB	(FSDE),GNWP
RPTO	DMPO
TSLB	(FSDE)
PDQ	(FSDE)
FLSW,WDL,TFML,WFL	(FSDE),NTLN
—	NULL,GSFD,PPTL,TRO
—	GTP,GDEC
ETR	TSM
—	GTP,GDEC
—	SVL,RTL
MVC,POT	TSTY
QPOP,DSPM	CR
IZTQ	—
WFD,GLBL	SFP,SNP,SPP,SMST
GFP,TFAT	—
TMZ	GFP
—	MTAP
VPAR,SMST	(FSDE)
—	NTLN,NLOC,VMDE

Routine	Location	Paragraph
VSMP	16510	4.4.7.13
WD	10462	4.4.2.4
WDCH	01313	4.4.1.25
WDL	02331	4.4.2.5
WEP	10210	4.4.1.26
WFD	00670	4.4.2.6
WFL	10450	4.4.2.7
WL	02275	4.4.2.8
WM	10043	4.4.6.3
WSC	10220	4.4.2.9
WXL	01752	4.4.2.10
X10	03720	4.4.9.28
ZLB	02313	4.4.3.4
ZSWR	01131	4.4.1.27
ZT	11400	4.4.9.29

Calls	Called By
SMST	(FSDE)
WFD	—
CKSW,WFD,SNL,GLBL	KBI
FLSW,WFD	WL,SVL,RTL
WSC	SEOP,MVC
—	ZLB,GM,WM,DSPM,WD,WXL,ERAS,MVC CKSW,TSLB,GLBL,WSC,WDCH,WDL,ATRK
CKSW	SVL,RTL
CKSW,WDL	BLKL,EL,MVC
CKSW,FLSW,WFD	DSPM
CKSW,WFD	WEP
WFD	ERAS
—	GDEC
SNL,WFD,BLNB	MVC
—	ERAS
—	(FSDE)

P.3 Source Handling Routines

Name	Location	Calls
ATRK	013600	WFD,SMST,CKC
CLLN	013524	
CNSL	012461	IDPT,TFIO,GPSC,PDL,COLB
COLB	012254	CLLN
CSH	013660	SHP, FKBS,CKC
CSL	012446	LDP,CNSL
DPAR	013312	FDL,DTB
DSL	012100	LDP,DSPS
DSPS	012117	IDPT,TDL,TFEI,TFIO,GPSC,FDL,TFEO,PDL,COLB
DTB	013371	X10
EATK	013613	SMST,LBSX,CSH
ECSL	013044	LBSX,CSL
EDSL	013033	LBSX,DSL
EGSD	013011	LBSX,GSFD
EPSD	013022	LBSX,PSOD
ETTK	013000	LBSX,TTK
FDL	013273	
GPAR	013135	EAK,FDL,GSTR,DPAR,FKBS
GPSC	012401	SOBC
GPTL	012642	
GPW	013224	FDL
GSFD	012544	NULL,SWR,LUPI,TDL,PPTL,ATRK,CKC
GSTR	013203	GPW
GTWP	013500	GPAR

Called By

TFIO,PSOD,GSFD

COLB

CSL

DSPS,IDPT,CNSL,PSOD

EATK

ECSL

GPAR

EDSL

DSL

DPAR

(FSDE)

(FSDE)

(FSDE)

(FSDE)

(FSDE)

(FSDE)

DSPS,TFEC,GPAR,GPW,DPAR

LDP,GTWP

DSPS,CNSL

PSOD

GSTR

EGSD

GPAR

LUPI,LUPO

Name	Location	Calls
IDPT	012204	COLB
LBSX	013571	SMST
LDP	013125	GPAR
LUPI	012664	GTWP
LUPO	012675	GTWP
PDL	012224	PTDL,RWDL
PPTL	012624	SWR
PSOD	012517	LUPO,TRO,COLB,GPTL,PDL,ATRK,CKC
PTDL	013554	
SOBC	012422	
TDL	012240	PTDL,RWDL
TFE	013537	TFES
TFEC	012333	FDL
TFEI	012264	TFE
TFEO	012440	TFE
TFES	012272	TFEC
TFIO	012350	ATRK,TDL,TFEI
TNTK	013072	RWI
TRO	013510	SWR
TTK	013055	LDP,TNTK
X10	013410	

Called By

DSPS,CNSL

ETTK,EPSD,EDSL,ECSL,EATK

DSL,CSL,TTK

GSFD

PSOD

DSPS,CNSL,PSOD

GSFD

EPSD

PDL,TDL

GPSC

DSPS,TFIO,GSFD

TFEI,TFEO

TFES

DSPS,TFIO

DSPS

TFE

DSPS,CNSL

TTK

PSOD

ETTK

DTB